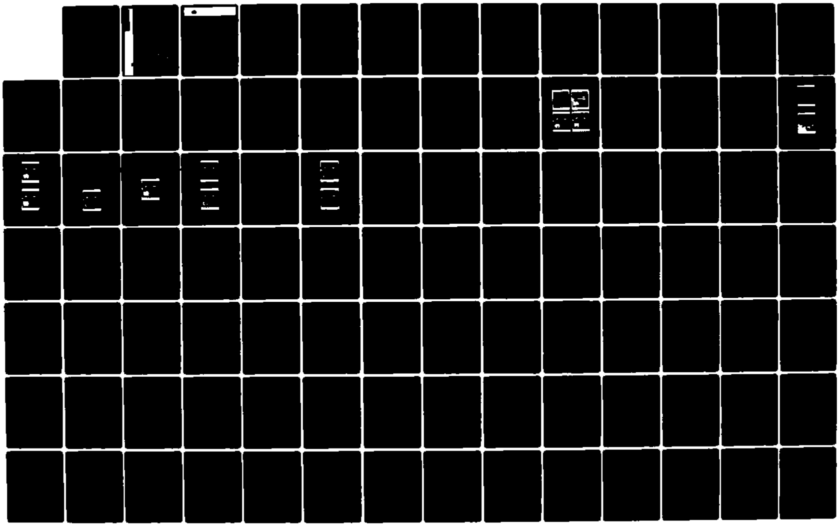
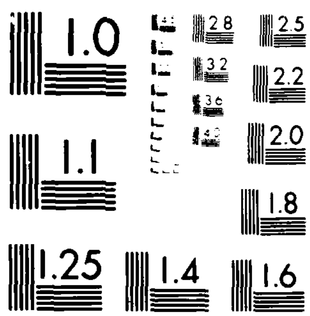
AD-A122 473 ARTIFICIAL INTELLIGENCE -- RESEARCH AND APPLICATIONS 1/3
(U) STANFORD RESEARCH INST MENLO PARK CALIF B RAPHAEL
JUN 76 DAHC04-75-C-0005

UNCLASSIFIED F/G 6/4 NL





MICROCOPY RESOLUTION TEST CHART

1

Annual Progress Report *June 1976*
Covering the Period 1 April 1975 through 11 April 1976

ARTIFICIAL INTELLIGENCE -- RESEARCH AND APPLICATIONS

Prepared for:

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY
ARLINGTON, VIRGINIA 22209

SRI Projects 3805 and 4763

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

STANFORD RESEARCH INSTITUTE
Menlo Park, California 94025 · U.S.A.

Annual Progress Report June 1976

Covering the Period 1 April 1975 through 11 April 1976
Stanford Research Institute Projects 3805 and 4763

ARTIFICIAL INTELLIGENCE -- RESEARCH AND APPLICATIONS

Edited by

Bertram Raphael

With Contributions by Members of the Project Staff

Contract DAHC04-75-C-0005	Effective Date of Contract: 10 October 1974
ARPA Order Number 2894	Contract Expiration Date: 9 October 1975
Program Code Number 61101E	Amount of Contract: $750,837
Contract DAAG29-76-C-0012	Effective Date of Contract: 10 October 1975
ARPA Order Number 2894	Contract Expiration Date: 6 June 1976
Program Code Number 61101E	Amount of Contract: $349,333

Prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

ARLINGTON, VIRGINIA 22209

ABSTRACT

This report describes the work performed during a transition year of research and applications in several aspects of artificial intelligence.

Work on a "computer-based consultant" was terminated. Aspects of that project that were completed this year and are briefly documented here include the "inference net" approach to developing a practical Bayesian updating procedure for rule-based systems; the QLISP programming system; and a system for locating objects in multi-sensory images. Other interim studies reported here include an overview of the state of technology in artificial intelligence, including a bibliography of 65 references; a detailed plan for a three-year R&D program leading to a computer-based consultant system for military vehicle maintenance; and a report outlining how sensors and prognostic methods can play an important role in vehicle maintenance.

A major section of this report describes the first six months of a major new effort in support of the ARPA/IPTO "Command-Control Architectural Testbed" program. We investigated the data management systems available on the ARPAnet, especially the Datacomputer software; created on the Datacomputer a data base about 214 ships, which is now available over the net to interested researchers; developed and implemented a preliminary version of a File Access Manager for robust access to a distributed data environment; adapted existing speech-understanding software to process textual input; applied an SRI-developed language interface package to the specific task of processing queries to that data base; and demonstrated an initial system that answers English questions about the contents of a remote data base in real time.

The last section of the report deals with another new area: Cartography and Photo Interpretation. A survey of current practices and problems in these domains helped define research goals. Preliminary experiments show the feasibility of applying AI tools and approaches to achieve these goals.

CONTENTS

APPENDICES

ILLUSTRATIONS

SECTION II:

SECTION III:

SECTION V:

I INTRODUCTION

This report describes the activities of the SRI Artificial Intelligence Center that were supported by ARPA under the cited contracts during the year from April 1975 to April 1976. This was a transition year for our activities under ARPA support; during this period, at ARPA's request, we terminated one major program (the Computer-Based Consultant effort), concluded some related technical studies, conducted a number of interim investigations, and began work on two major new programs that will form the core of our future ARPA activities.

During the period from October 1973 through March 1975 we worked on a system called the Computer-Based Consultant. This work, which was viewed as the initial part of a five-year effort, was aimed at developing a system that would be able to talk (in natural language) with a human user to help him or her perform tasks in some particular task environment. We intended to build an automatic consultant that approximated the communication, perceptual reasoning, and factual knowledge skills of an actual expert on the scene. Our main goal was to create the fundamental technology needed to build such consultant systems, with the expectation that a good portion of this technology would be independent of the details of the particular task environment. Thus the work had potential high payoff because of the variety of applications in several task environments in which consulting expertise is needed or would be helpful.

Although the Computer-Based Consultant effort was prematurely terminated early in the period covered by this report, several of its principal elements had already evolved to the point where they promise to have independent value. We therefore continued work on these elements so that they could be clearly demonstrated and documented.

Section II of this report briefly describes the results of completing these separate efforts: the "inference net" system for representing and combining probabilistic, rule-based knowledge; the QLISP programming system; some experiments with a planning system for visual image analysis; and a broader look at the nature of the planning process in automatic problem-solving systems. Since each of these topics has already been documented in considerable detail in technical notes or papers, Section II of this report is limited to overview summaries, with reference citations to more-detailed documentation.

One of our major activities during the first half of the year covered by this report was to review the state of the technology of artificial intelligence in general and of SRI's artificial intelligence capabilities in particular, with the aim of helping ARPA decide upon future directions for work in this field. Section III of this report contains some of the results of these reviews. Section IIIA is a broad overview of the AI field. Section IIIB contains a detailed plan for how a computer-based consultant system could be developed and applied specifically to help maintain military vehicles such as jeeps. This consideration of vehicle maintenance led us to suggest the possibility of augmenting an AI approach with on-board sensors and prognostic methods for improving the efficiency of a maintenance program. Section IIIC contains preliminary suggestions for the use of prognostics in vehicle maintenance.

Six months ago, ARPA chose not to continue supporting the computer-based consultant or vehicle-maintenance programs at SRI. Instead, two new program areas were selected by mutual agreement. The last two sections of this report present the results of our initial six-months' efforts in these areas: Section IV presents our work on decision aids for command and control, which is in direct support of the ARPA/IPTO "Command-Control Architectural Testbed" program; Section V presents our work on interactive aids for cartography and photo interpretation, which is intended to be an element of ARPA's new program in "Image Understanding." Both these programs are expected to be multi-year

efforts, which will occupy a growing share of our resources in the future. We are pleased with the progress that has already been achieved, as documented in Sections IV and V of this report, in a brief start-up period.

Previous annual reports have each included a section about changes to our PDP-10 computer facility. No such section is included here, because the facility has not changed significantly during the past year--except for a steady increase in user load, and corresponding decrease in resulting responsiveness. The facility has become one of the most reliable and most heavily used on the ARPAnet. Since its usage has been monitored on a continuing basis by ARPA/IPTO, no special report seems necessary at this time. During the next year we plan a major change in our computing arrangements, which will result in the availability of increased computing power at relatively lower cost.

Because of the diverse natures of the various sections of this report, we have chosen to give each section a high degree of integrity. Therefore the illustrations are numbered separately in each section, and the references and appropriate appendices appear at the end of each section rather than together at the end of the entire report.

II CONCLUSIONS OF PREVIOUS TASKS

A. Bayesian Updating and Inference Networks
by Richard O. Duda

In our previous report we described a rule-based, hypothesis-and-test approach to the diagnosis of mechanical equipment [1]. In this section we review this general approach and summarize our accomplishments in this area during the last year. Details concerning the theoretical analysis and the computer implementation can be found in [2] and [3], respectively.

With a rule-based diagnosis system, relations between symptoms and fault hypotheses are represented as production rules, much as was done in the MYCIN program for medical diagnosis [4]. Associated with any hypothesis is a probability that the hypothesis is true. Discovery of evidence, whether volunteered by the user or obtained at the request of the system, modifies the probability of one or several hypotheses. These new probabilities determine the next test requested by the system. Hopefully this sequential process of gathering evidence and updating hypotheses converges with one very likely hypothesis that explains the fault at the current level of detail.

In general, the rules linking evidence and hypotheses can be represented as a graph. Because a confirmed hypothesis can also serve as evidence for a higher-level hypothesis, both evidence and hypotheses are represented as nodes in the graph,, the rules being arcs linking the nodes. This abstract representation is applicable not only to diagnosis but also to other inference problems characterized by incomplete and/or uncertain information. We have called such a graphical representation an inference network, or an inference net for short.

One of the important problems that arise with inference nets concerns the propagation of probabilities. Initially, each node is assigned a prior probability of being true, and each arc is assigned two conditional probabilities--one giving the probability of observing the evidence when the hypothesis is true, and one giving the probability of observing the evidence with the hypothesis is false. When a piece of evidence is obtained, the probability of the corresponding evidence node changes. Bayes' rule provides the basis for updating the probability by that evidence [1]. However, several problems arise that complicate a straightforward application of Bayes' rule:

(1) The evidence may not be known to be true or false with certainty, but only to some degree. This is particularly true for nodes whose probabilities have been indirectly established through a chain of one or more inference rules.

(2) Several pieces of evidence may bear on one hypothesis, and they may not be independent. This is particularly obvious when subsumption exists. For example, if we have the rules E1 - H and E1 ^ E2 - H, it is obvious that the truth of E1 ^ E2 is not independent of the truth of E1.

(3) Constraints may exist among hypotheses that may be inconvenient to represent with the network formalism. For example, if n top-level hypotheses are mutually exclusive and exhaustive, then any evidence that lowers the probability of one hypothesis should raise the probabilities of the other n-1 hypotheses. However, when n is large it is uneconomical to have the n-1 rules that would accomplish this slight adjustment explicitly present.

(4) Rules of a general nature, such as "a failure of an electrical system to function suggest lack of proper input signals or power" should be represented once rather than being replicated many times. This leads to rules containing variables that must be matched to situations at run time, and to problems involving time/space tradeoffs.

(5) The prior and the conditional probabilities are estimated values obtained by interviewing experts. As a consequence, they are usually not consistent, and these inconsistencies have been observed to cause serious errors in the propagation of probabilities.

Substantial progress has been made on solving these problems and developing a practical Bayesian updating procedure for rule-based inference systems. The problem of using uncertain evidence in a way

that tolerates inconsistencies has been largely solved, as is described in detail in [2]. A system called INET [3] that incorporates this updating procedure has been implemented. INET uses a heuristic path-tracing procedure to discover and correct for certain kinds of subsumption, and has used a normalization procedure to handle the constraint of mutually exclusive and exhaustive hypotheses [5]. Although developed with mechanical diagnosis in mind, this approach is quite general, and is currently being seriously considered for such diverse applications as to certain problems in electronic warfare [5], intelligent terminals [6], mineral exploration, and agricultural management.

B. QLISP

by B. Michael Wilber

During the past year we packaged QLISP as a finished project. This included publishing a reference manual for the system [7] as well as building an export version of it. Both the manual and the source files are available from the Artificial Intelligence Center.

The manual is primarily a reference for QLISP which attempts to serve as well as an introduction to the language. Since QLISP programming diverges strongly from traditional languages, the manual also describes the new concepts for which QLISP was implemented. The implementation of QLISP is almost as smooth from a user's viewpoint as the implementation of INTERLISP [8]. Only the tremendous flexibility of INTERLISP permitted this; nevertheless, QLISP needed description as a programming system supporting the programming language, and that description is also included in the manual.

We also have available an export version of QLISP. The symbolic source files are up to date and available from us. These source files are sufficient to build a QLISP from scratch. In fact, even previous revisions of these files have given only minor trouble to people building their own copies of QLISP.

Thus we now have available a complete, self-contained export version of QLISP. With it, anybody should be able to use QLISP without guidance or advice from SRI or reference to SRI.

C. Experiments with a System for Locating Objects in Multi-Sensory Images*

by T. D. Garvey

1. Introduction

This section [9] describes a goal-directed perception system, which is described in detail in Reference [10], that locates objects in images of rooms by planning and executing special purpose strategies. These strategies use various kinds of knowledge including object descriptions, world models, and sensor models, to determine those features which distinguish the target object from other known objects. This planning approach allows the system to confront directly the problem of sensory overload, by using only that data required for the task. The plans flexibly integrate data from multiple sensory modalities, and take advantage of natural contextual constraints. Distinguishing features strategies have been developed and executed to find many objects in several different scenes; in this paper we describe one experiment where the system must deal with an unexpected object. This experiment illustrates a number of important points: the overall operation of the system, the options the system has at various stages of execution, the results obtained from the execution of plan steps, and the effect of an unexpected object on the plan.

Before discussing the example, we provide a brief overview of the system and establish the experimental context, including the primitive operators available, equipment used, the data available from the equipment, and object representations.

----

* A slightly revised version of this section has been accepted for presentation at the International Joint Conference on Pattern Recognition in November, 1976.

a. Overview

A plan is an AND/OR graph (with loops) whose "tip" nodes are program modules that, when executed, examine either image data or a symbolic data base in order to determine various types of requested information. For example, a module might be designed to locate all the sample points in a set whose brightness is between two limits. The module reports success if it can satisfy the request, and failure otherwise. In our example, the module would report success if it found any samples with the appropriate brightness. such modules are called "executable subgoals", and executing a subgoal means running the program module.

The system computes a plan for locating an object based on the paradigm of acquisition, validation and bounding. The goal of acquisition is to select a set of image points that are likely to belong to the target object; validation is intended to eliminate points from other objects that share the acquisition attributes of the target object, thereby increasing the likelihood that the points remaining really are from the target, and the bounding phase is designed to extract the boundary of the object. Acquisition will typically require the application of a detector to selected image points to determine whether they belong to a given object. Detectors are generated automatically for particular objects.

Plans usually have options at each step, and a scoring routine is used to determine what to do next. The scoring routine looks at each plan step (or subgoal) and computes a score based on the cost (in time) of execution of the step, and the likelihood (or confidence) of success. These scores are passed from executable goals through intermediate steps to the initial goal. By following back from the top node downward, the path leading to the best subgoal to be executed can be selected.

After the best executable subgoal is selected and executed, the system determines whether the main goal was achieved

(i.e., the object has been located), whether it could not ever be achieved (i.e., there are no further options remaining), or whether another iteration of the sequence is required. The process continues until success or failure of the main goal.

b. Operators

Detectors

The detectors used in these experiments are indicated in Table 1. The measured attributes are brightness, hue, saturation, height and local surface orientation. Brightness is measured to 32 gray levels. Hue (or color) is measured as an angular displacement around the center of a standard color triangle (i.e., around the "white point"), with red being represented by an angle of zero degrees, green by an angle of 120 degrees, and blue by an angle of 240 degrees. Saturation (color purity) is measured from 0 to 1, with 1 representing a pure color and 0 representing a shade of gray. Height (Z) is measured in inches above the floor, and orientation is measured in degrees from the Z axis. These measurements are discussed in greater detail in [10].

The table gives the average cost (in milliseconds) of measuring the value for a single sample.* A detector is simply a predicate that determines whether a measured attribute value lies within a specified interval. The detector returns true when the value is within the limits, false otherwise.

Detector	Cost
BGHT (brightness)	76
HUE	101
SAT (saturation)	95
RANGE	99
ORIENT	110
EDGEP (edge op)	70

Table 1. Available Detectors

----

* A sample generally implies a single point, although for computing orientation, it will mean a small patch of points.

Cost is normally defined as the expected execution time required to take the measurement. For these examples, we have used cost values for range derived attributes that are far below actual current costs. Taking a range finder picture is a lengthy process. Our present system uses a low-power laser, and typically requires almost three hours to scan a scene at 128 by 128 resolution. Since we expect the whole process (which is still experimental) to be greatly improved, we elected to compute cost as the average time to take the range measurement (not including settling time) plus the time requested for any associated computation.*

Primitive Operators

To extract information from an image, the system has a number of primitive programs available. These are listed in Table 2, categorized by function. Briefly, the system can use FILTER-WINDOW and SCAN for acquisition. FILTER-WINDOW examines a set of points in a window of the picture, and returns all those that are accepted by a particular detector. SCAN-UP and SCAN-DOWN require a previously located object for a starting point, and sequentially examine points on an appropriate line segment, searching for the target object.

Acquisition	Validation	Bounding
FILTER-WINDOW	VALIDATE	HRB
SCAN-UP	DISTINGUISH	VRB
SCAN-DOWN		GROW-REGION

Table 2. Program Primitives

To verify acquisition, the system uses DISTINGUISH or VALIDATE. DISTINGUISH classifies each point as belonging to one of the objects that might be ambiguous with respect to the acquisition detector. If a point is classified as the target object with sufficient likelihood (currently the classification likelihood must be greater than

----

* In addition, it is frequently the case that alternative means of obtaining the information are either not available or equally costly.

.8), the point is retained, otherwise it is deleted from the set. VALIDATE checks remaining (i.e. unchecked) attributes to ensure that their values fall within range for the object.

After the object has been located, the system can compute its outline in a variety of ways. The horizontal or vertical rectangle bounders (HRB and VRB) are useful for appropriately shaped objects. These programs scan for an edge of the object (using the edge operator), predict and locate the perpendicular edges, and then locate the last edge of the boundary. Alternatively, the system may elect to use GROW-REGION to extend the initially acquired points to the boundary, and then generate the boundary with the convex hull routine, HULL.*

c. Relationships

The system knows about a small number of object relationships, including ABOVE, BELOW, BEHIND, IN-FRONT-OF, SUPPORTS, SUPPORTED-BY, AND ADJACENT.** Relationships between objects are supplied by the user, and not computed by the system; their main use is in planning strategies.

Associated with the relationships are the likelihoods that the relation holds. A typical relationship is TELEPHONE SUPPORTED-BY TABLE. Given that TELEPHONE is in the scene, the probability that it is supported by TABLE is supplied by the likelihood on the relationship.

These relations serve two particular purposes in our system. First, they are used to decide if an object is pictorially adjacent to another. That is, whether the objects are adjacent in the image. This is useful for SCAN and GROW, both of which normally need to know what objects are adjacent to the one they are interested in.

----

* The use of HULL, while not always appropriate, is motivated by the fact that most of the objects in our experimental world are convex, and HULL provides a simple, cheap way of generating this outline (see Appendix 2).

** BEHIND, SUPPORTS, and SUPORTED-BY imply ADJACENT.

Relations are also used to compute windows for an object provided by another object. For example, after finding the table, a window consisting of a volume of space above the table can be computed. This window then constrains the location of objects likely to be on the table top. From this window, the system also computes the set of objects likely to be in the window. For example, the window associated with the table top contains all objects likely to be on the table top, plus such pictorially adjacent things as the wall.

These relations are quite simple, and designed to provide useful information from a scene with a certain "normal" viewing position. If the scene were viewed from another angle, they would not be sufficient. There is no conceptual reason for not using a model which could be geometrically transformed to account for any viewing angle. However, this was more complicated and would have been a diversion, without significantly changing the essence of our work.

d. Objects

Table 3 lists a few of approximately fifteen objects known to the system when these examples were run. The attribute values listed represent the extremes of those values, for each object. The system actually makes its choice of detectors based on histograms of attribute values for each object. Relations and associated likelihoods are also listed for each object.

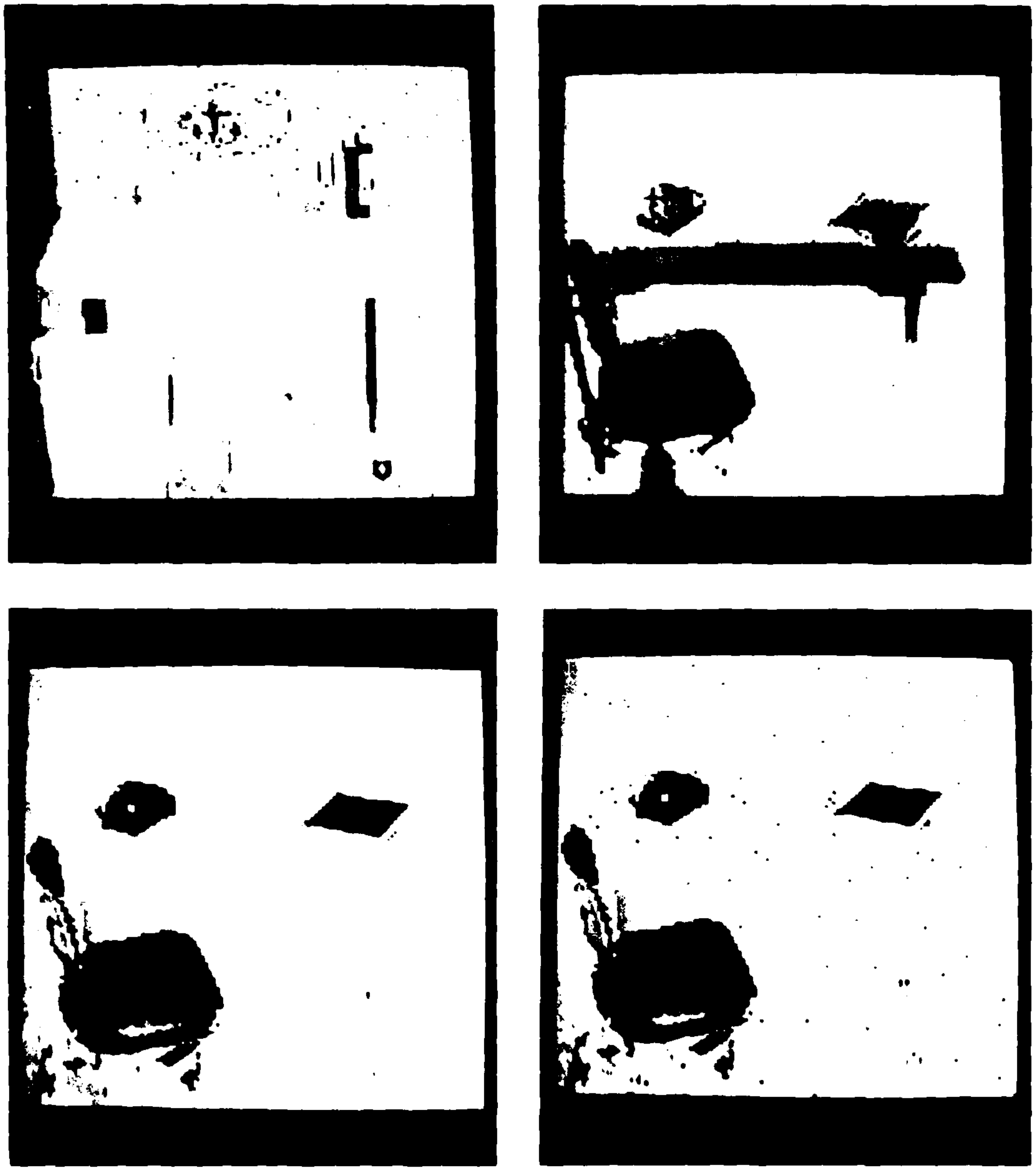
These object characterizations were generated in an interactive session with the system. The objects were indicated to the system (in various images), by manually outlining them on the display. The system then measured the attributes of the indicated regions, and computed the corresponding histograms. After computing these histograms, the system interrogated the user about related objects, and stored all relationships specified.

Object	Description	Relations
TABLETOP	BGHT: 18 - 26 HUE: 26. - 58. SAT: .23 - .32 HEIGHT: 26.0 - 27.5 ORIENT: -7.0 - 5.5	SUPPORTS TELEPHONE .6 SUPPORTS BOOK .4 IN-FRONT-OF WALL 1.0
TELEPHONE	BGHT: 4 - 8 HUE: 72. - 125. SAT: .15 - .22 HEIGHT: 5. - 6. ORIENT: 82. - 92.	SUPPORTED-BY TABLETOP 1.0 IN-FRONT-OF WALL 1.0
SEAT	BGHT: 7 - 12 HUE: 110. - 146. SAT: .42 - .47 HEIGHT: 14. - 15. ORIENT: -15. - 10.	BELOW BACK1 .4 BELOW BACK2 .6 IN-FRONT-OF WALL 1.0
BACK1	BGHT: 6 - 9 HUE: 115. - 130. SAT: .4 - .5 HEIGHT: 18. - 28. ORIENT: 75. - 90.	ABOVE SEAT 1.0 IN-FRONT-OF WALL 1.0
BACK2	BGHT: 8 - 14 HUE: 75. - 240. SAT: .24 - .31 HEIGHT: 18. - 28. ORIENT: 60. - 90.	ABOVE SEAT 1.0 IN-FRONT-OF WALL 1.0

Table 3. Partial List of Objects

e. Equipment

The set of images making up a complete picture are obtained from two devices, a television camera and a laser range finder. A set of three television images, taken through red, green and blue filters provide information required to compute the color components, hue and saturation. A black and white image is shown in Figure 1-a. This set is complemented by a range image of the same scene, composed of distance measurements from the range finder center to each image element and associated calibration matrices. It allows the system to compute the X,Y,Z (i.e., world) coordinates of each point in the image, providing height (Z) and local surface orientation.



SA-4973-11

FIGURE 1 TELEVISION AND RANGE FINDER IMAGES

The range finder [11] is a scanning device which measures the phase difference between the reflection of a modulated laser beam and a reference beam. This provides the time-of-flight information required to compute the distance to the reflecting point. Figure 1-b shows a range picture displayed so that near points appear darker, and more distant ones, lighter.

In addition to range, the instrument also measures the amplitude of the reflected signal, which after normalization,* provides an accurate measurement of reflectivity of the point at the laser wavelength** (see Figure 1-c). The amplitude image provides significant advantages over normal television images by virtue of its increased dynamic range and its perfect registration (with no shadows) with the range image. Consequently, the laser amplitude picture was used to compute brightness.

The two separate optical systems employed for the television and range pictures require that the images be registered. This is accomplished by transforming the television coordinate system into the coordinate system of the range finder (as shown in Figure 1-d), using the calibration matrices computed before taking the pictures.

As can be seen from Figure 1-a and Figure 1-c, the images are not perfectly registered. Since the centers of the two optical systems are separated by eighteen inches (with the objects about about ten feet away), there are parallax discrepancies and the viewing areas of the two devices are not exactly identical -- the white strip on the right side of the image is that part of the scene visible to the rangefinder, but not to the camera. We confine our attention to the overlapped area where both range and brightness information are available. In addition, since we are generally interested in surfaces, rather than isolated points, parallax errors and slight misregistrations

----

* This normalization, and others are described in [11].

** A He-Ne laser operating at .6328 microns.

are not a problem.*

We will illustrate the example with the amplitude picture, since it is more distinct and easier to interpret (in black and white) than the television picture. Points in the image will be indicated as small light or dark (depending on the background) squares. Regions will be shown as strongly outlined areas in the image.

Find the telephone by acquiring, validating, and bounding the telephone:

Acquire the telephone by direct or indirect means;

Acquire the telephone directly by filtering and distinguishing, or,

Acquire the telephone indirectly by finding the tabletop, windowing and filtering within the window;

Find the tabletop by acquiring, validating, and bounding the tabletop;

Acquire the tabletop ...

When the plan is scored, the best score is provided by the path through "Acquire the telephone indirectly" to "Acquire the tabletop directly" to "Sample the scene at a density of .01 and Filter the scene with the predicate: (HEIGHT 27.0 28.0)** and distinguish the table top from objects: wall, books, telephone, and tapeholder." This plan recognizes the advisability of locating the telephone by first locating the tabletop, and then looking for the telephone on the tabletop. This approach is advantageous for two reasons: first, it is likely to be much cheaper to search the limited area of the tabletop for the telephone, rather than the whole scene (realizing that the tabletop is much easier to find than the telephone). In addition, the likelihood

----

* Even when the program filters point sets, there are generally several points selected from any given surface. By looking at these, the average characteristics of the point set tend to dominate, and slight discrepancies will tend to cancel.

** This detector, which is a simplified reresentation of a LISP program, requires the HEIGHT of a point to be between 27 and 28 inches.

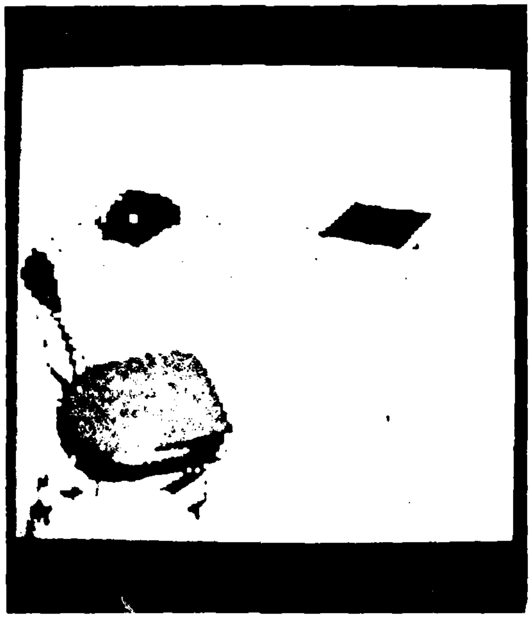
of confusing the telephone with other objects is much smaller on the tabletop than in the complete scene, since the system knows of no other objects with similar characteristics that are likely to be on the tabletop (but, of course, there may be unanticipated objects). It is important to keep in mind the fact that should the indirect approach fail, the system can always fall back on the direct approach. In addition, the detector for tabletop is taking advantage of its main characteristics, that it is a collection of points at a particular height.

In Figure 2, the system has sampled the scene at the prescribed density. These samples are filtered with the indicated height detector, resulting in the set of points shown in Figure 3. The program has also selected a number of points from other objects. These objects were known to be ambiguous with respect to the detector, and an appropriate step to distinguish them from the tabletop was included in the plan. After distinguishing the tabletop points from the imposters (using (AND (ORIENT -5.0 5.0) (BGHT 19 24))), as shown in Figure 4, the system progresses to the bounding phase of the strategy for locating the tabletop.

This step provides a choice between using the horizontal rectangle bounder (HRB) program or region growing. The likelihood of either working correctly is fairly large (close to 1.0), with the region grower slightly more likely to succeed. The real choice here depends on the expected cost.

Due to the work involved in the HRB program, the system opts for region growing. A region is grown outward from the boundary of the initial set of points. Since the set is close to the expected region in size, the system anticipates little extra work to locate the boundary.

The results of applying the region grower, using the predicate, (AND (HEIGHT 27.0 28.0) (ORIENT -5.0 5.0)), to check the height and orientation at each point, and then computing a convex hull,

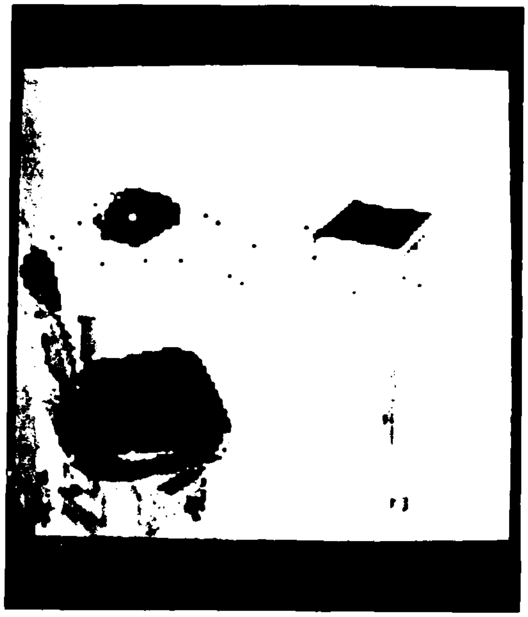


SA-4973-1

FIGURE 2 SCENE SAMPLED FOR TABLE TOP

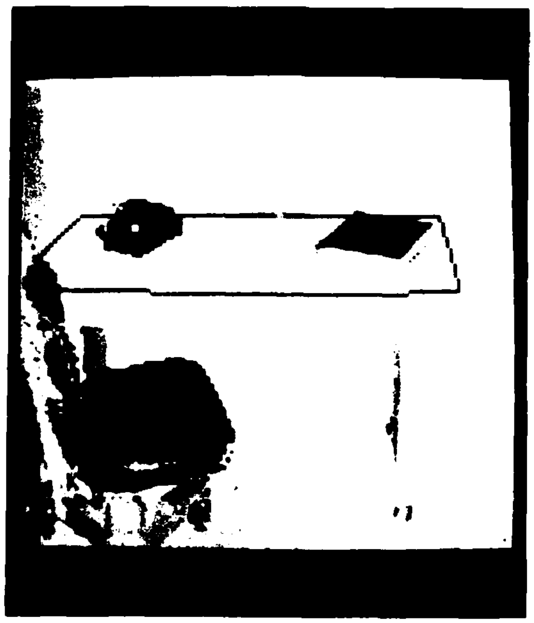
SA-4973-2

FIGURE 3 POINTS AT HEIGHT OF TABLE TOP



SA-4973-3

FIGURE 4 VERIFIED TABLE TOP POINTS

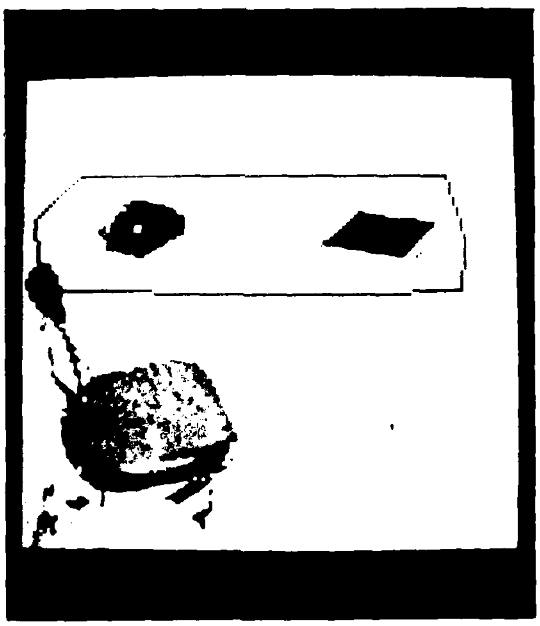


SA-4973-4

FIGURE 5 TABLE TOP OUTLINE

are shown in Figure 5. The system is now ready to compute a window for locating the telephone.

The window is computed from the boundary of the object that provides the constraint (in this case, the tabletop), the direction in which the window extends, the spacing between the region and the window, and the extent of the window. In particular, for locating the telephone, a window is computed upward from the tabletop, with a spacing of zero (since the two objects are adjacent), and extending approximately six inches above the tabletop. The window's projection in the image is computed by finding the X,Y,Z coordinates of the window vertices, and then computing their projections in the image. The region of the scene so defined is shown in Figure 6 overlaid on the defining region to emphasize the derivation. Another important piece of information is the set of objects that might be found within the window. This set is computed at planning time, and consists of all other objects having the same relationship to tabletop as telephone (i.e. SUPPORTED-BY).



SA-4973-5

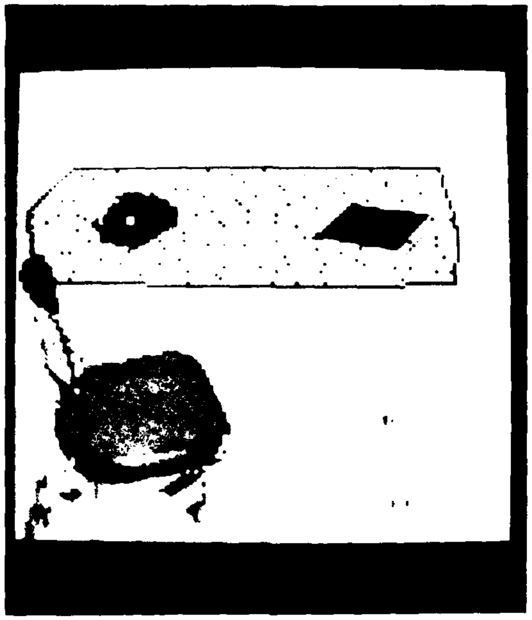
FIGURE 6 PROJECTED WINDOW FOR TELEPHONE

The next step in the plan is to filter the new window for telephone points. The window is sampled, as shown in Figure 7, and filtered selecting all points passing the brightness detector (B&W). This is shown (in white) in Figure 8. Based on its current knowledge of objects in the environment, the system has mistaken points from the notebook as telephone. In the course of validation, the system is unable to detect its error, and retains all the acquired points. It does notice that the area covered by the points is greater than the size of a telephone, and splits the region, treating the two new regions as if they were each possible telephones. In Figure 9 the system grows the regions, and finishes by identifying them as telephone.

SA-4973-6

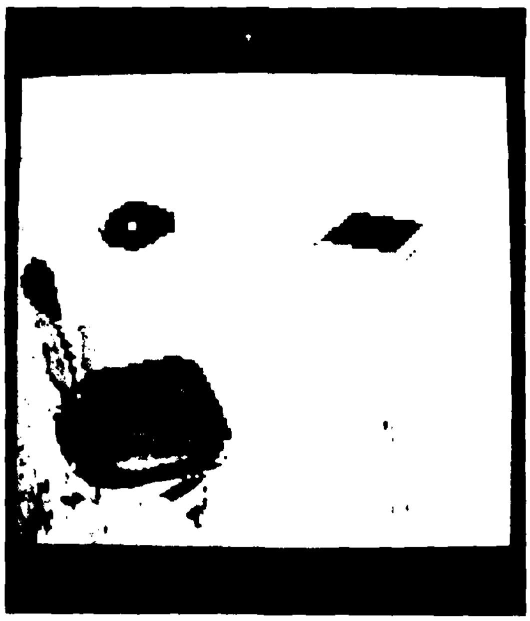
FIGURE 7 SAMPLE WINDOW FOR TELEPHONE

Although the result was incorrect, it really was not in error. The system produced the best plan for the available world knowledge. Therefore, to improve the results, the system needs additional knowledge about notebooks. The first step in teaching the system about notebooks is to indicate the region of the image occupied



SA-4973-7

FIGURE 8 INITIAL TELEPHONE ACQUISITION POINTS WITH ERRORS



SA-4973-8

FIGURE 9 REGIONS IDENTIFIED AS TELEPHONE

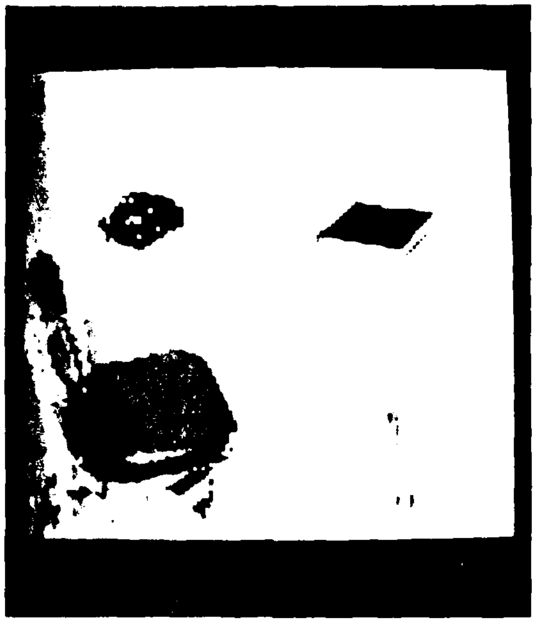
by the notebook to the program, and to supply the appropriate relationships, allowing it to characterize the notebook, exactly as was originally done for the initial set of objects.

Now, armed with a fresh description of a notebook, the system is again requested to locate the telephone. The plan is virtually the same as before (the main difference in the overall plan is to make the direct acquisition of the telephone even less attractive, since now the detector for telephone must also discriminate against notebooks, thus requiring even longer execution times). The plan is executed up to the point where the window provided by the tabletop is created and sampled.

This time, the system is aware that notebooks are likely to be in the window, and generates the acquisition predicate, (AND (BGHT 4 7) (ORIENT 60.0 90.0)), to guard against making another mistake. Since the notebook example it has seen had a horizontal surface orientation, the detector for telephone requires points to be off the horizontal. This test effectively eliminates all but points on the telephone as shown in Figure 10. The final result is again grown out to produce the correct telephone outline in Figure 11.

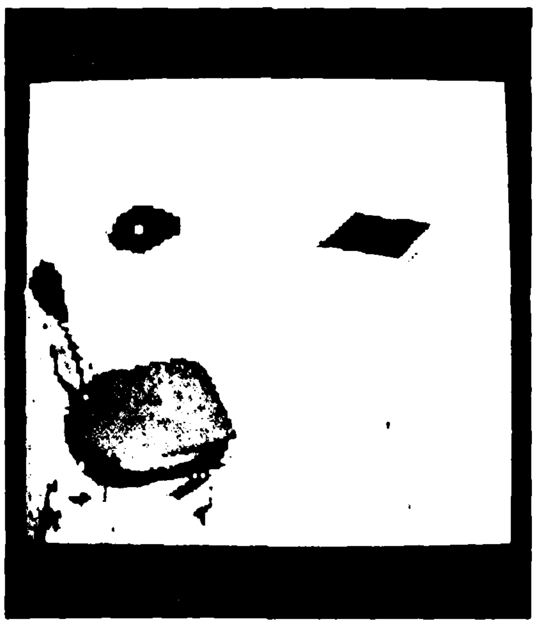
2. Summary

This work was based on the concept of "vision by distinguishing features." Object recognition via distinguishing features is performed by looking only for those features that differentiate the object from other known objects in a particular context, thereby allowing most of the image to be quickly eliminated from consideration. Only those areas that have the distinguishing characteristic need to be processed further. By planning a perception strategy as needed, using information available to the system at the time, the system easily accepts new objects and is easily extendable to new sensory modalities. The strategies allow the system to confront sensory overload by dynamically ordering the feature extraction



SA-4973-9

FIGURE 10 CORRECTLY ACQUIRED TELEPHONE POINTS



SA-4973-10

FIGURE 11 CORRECT TELEPHONE OUTLINE

operators to check simple features (in the current context) first, and more expensive features later, after the context has been restricted sufficiently to warrant their use.

The error committed by the system in identifying the notebook as a telephone is the type apt to be committed by any system which must rely on a set of rules and information which are discovered to be insufficient to the task. The capability to plan new strategies, however, provides the crucial capability to incorporate new knowledge so quickly and gracefully. This incremental acquisition of knowledge is very important to a an intelligent system. It allows the system itself to demonstrate what information it needs, and only that information needs to be added; the system can then take full advantage of new information.

D. Planning Systems

by Richard E. Fikes

An automatic planning capability was an important part of the design of the CBC system. In order to synthesize some of the ideas developed for doing automatic planning during the project and to provide an indication of how those ideas relate to other similar systems, we conducted an investigation of automatic planning systems during this period and produced a survey paper containing the results of the investigation [13]. We focused particularly on facilities in planners for representing various kinds of knowledge because we felt that the effectiveness of such systems depends, to a large extent, on their ability to make use of descriptions and expertise associated with particular task domains. Examples of such domain knowledge include action models, state description models, scenarios, and special purpose plan composition methods. The paragraphs below present a brief summary of the survey.

A planner is needed when there is no prestored method for accomplishing a particular task. A task is usually specified to a

planner by describing an initial situation and a desired (goal) situation. The system is aware of a collection of actions (methods) that the active agent (e.g., a mechanical manipulator or a human to which the system is providing instructions) can perform, and the plan it produces is a sequence of these actions that are expected to transform the initial situation into the desired situation. Hence, the planner's role is to combine the methods that are available to the system in order to produce a new "method" that will accomplish the task. The methods being composed may consist of single actions or of multiaction scenarios, and the composition process may include instantiating and modifying the existing methods to match the particular situation.

Typically, a planner proceeds by hypothesizing sequences of actions for inclusion in a plan and testing each hypothesis by simulating the action sequence. The simulation produces a description of the situation that would be expected from the actions, and that situation is examined to determine if it satisfies the subgoals that the planner is trying to achieve. The basic method for creating the hypothesized action sequences applies means-ends analysis to determine "relevant" actions and to decompose the original goal into appropriate subgoals.

In addition to the basic simulation and means-ends analysis facilities, planners have been augmented so that they can easily accept planning expertise specific to the particular domain in which tasks are to be given. That is, in any domain there will be planning strategies and heuristics that a planner designed specifically for that domain would employ. Examples of the forms in which such information is included in planners include (a) procedural action models that incorporate planning strategy information about how to achieve the action's preconditions, (b) subplanners specifically designed for specialized classes of tasks, and (c) procedural inference rules embodying semantic properties of the relations being used to describe situations.

The power of planning systems has also been increased by providing facilities for constructing plans hierarchically. In this style of planning, a complete plan is made at a very abstract level in which much of the task's detail has been suppressed. Each step in this high-level plan can then be expanded into a slightly less abstract plan, and so on until a plan is produced at the desired level of detail. This approach to planning has been found to provide significant advantages during both the generation and the execution of plans.

We concluded that automatic planning facilities are an important component of many intelligent systems, and that a useful and effective technology has been developed for providing such facilities.

REFERENCES

1. N. J. Nilsson, ed., "Artificial Intelligence--Research and Applications," Progress Report, SRI Project 3805, Contract DAHC04-75-C-0005, Stanford Research Institute, Menlo Park, California (May 1975).

2. R. O. Duda, P. E. Hart, and N. J. Nilsson, "Subjective Bayesian Methods for Rule-Based Inference Systems," Proc. National Computer Conference (New York, 1976). Artificial Intelligence Center Technical Note 124, Stanford Research Institute, Menlo Park, California (January 1976).

3. G. Sutherland, "Implementation Inference Nets--II," Technical Note 122, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (January 1976).

4. E. H. Shortliffe, "MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection," Stanford Artificial Intelligence Laboratory Memo AIM-251, Stanford University, Stanford, California (October 1974).

5. W. F. Rousseau, T. D. Garvey, and H. S. Hewitt, "Application of Machine Intelligence to Electronic Warfare," Final Report, SRI Project 3372, Contract F33615-75-C-1306, Stanford Research Institute, Menlo Park, California (in preparation).

6. J. J. Gillogly, personal communication concerning the Rand Intelligent Terminal Agent, under development for the Defense Advanced Research Projects Agency under Contract DAHC15-73-C-0181.

7. B. M. Wilber, "A QLISP Reference Manual," Technical Note 118, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 11, 1976).

8. E. D. Sacerdoti et al., "QLISP: A Language for the Interactive Development of Complex Systems," Artificial Intelligence Center Technical Note 120, Stanford Research Institute, Menlo Park, California (March 1976).

9. Teitelman, Warren, INTERLISP Reference Manual. XEROX Palo Alto Research Center, Palo Alto, California 1974.

10. T. D. Garvey and J. M. Tenenbaum, "On the Automatic Generation of Programs for Locating Objects in Office Scenes," Proc. Second International Joint Conference on Pattern Recognition, pp. 162-168, Copenhagen, Denmark (August 1974).

11. T. D. Garvey, "Perceptual Strategies for Purposive Vision," Technical Note 127, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (April 1976).

12. D. Nitzan, A. E. Brain, and R. O. Duda, "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis," Technical Note 128, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (March 1976).

13. R. E. Fikes, "Knowledge Representation in Automatic Planning Systems," Technical Note 119, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (January 1976).

III INTERIM STUDIES

A. State of Technology in Artificial Intelligence*
by R. O. Duda and N. J. Nilsson

1. The Nature of Artificial Intelligence

a. Short History

Soon after large computers became available, scientists began attempts to use them for purposes other than routine numerical computations. Allen Newell, now at Carnegie Mellon University, was one of the early pioneers who showed that computers could be used to process symbolic data as well as numerical data. Workers in the late 1950s and early 1960s wrote programs that solved simple puzzles, proved theorems in logic and geometry, performed symbolic mathematical operations such as indefinite integration, and played games such as checkers and chess. These programs were the beginning efforts in a new subspecialty of computer science: Artificial Intelligence (AI). Many of the foundation ideas in AI, such as the heuristic search process, were solidified during this early period. In addition, special tools such as the list processing languages IPL and LISP were developed. (For a representative view of some of this early work, see Feigenbaum and Feldman, 1963.)**

Toward the end of this period, and signalling the beginning of the next, AI research groups were formally instituted at MIT and then at Stanford University. Together with the group at Carnegie, these groups began a more systematic attack on certain AI research problems involving natural language understanding, automatic problem solving, and visual scene analysis. It was during this second

----

* A modified version of this section has been submitted for separate publication in Wegner and Wulf (1976).

** References citations in this subsection refer to the Bibliography that follows this section.

period (about 1963) that funding of AI research was initiated by the Advanced Research Projects Agency (ARPA).

Toward the late 1960s, another period began with some of the groups concentrating on the development of integrated robot systems. SRI joined the other three groups as a major AI research center at about this time. This period also saw the beginning of "expert problem solving systems"--systems that possessed a large amount of specific knowledge about a particular domain, such as symbolic integration (Moses, 1967) and mass spectrometry (Buchanan et al, 1969). Because of this emphasis on expert problem solving, a much clearer understanding was obtained of the relative roles of search, specific domain knowledge, and techniques for representing knowledge. In the early 1970s, the present or fourth period began--a period increasingly devoted to applications. A major application, supported by ARPA, is the attempt to build systems to understand continuous speech (Newell et al., 1971). Other important applications efforts were begun or refined in chemistry (Sridharan, 1971; Buchanan and Lederberg, 1971), medicine (Kulikowski and Weiss, 1972; Shortliffe et al., 1973), mathematical symbol manipulation (Martin and Fateman, 1971), and automated manufacturing (Rosen, 1976). Various high-level computer languages for AI research were also developed during this period. (See Bobrow and Raphael, 1974.)

The current emphasis on applications ought to be regarded as initial attempts to apply a recently developed body of technology to significant practical problems. Much is being learned by these attempts that strengthen the technology. The technology has not yet settled into a "handbook" state, but the current period ought to see much of it stabilize.

b. The Major Participants

The four major American groups--Carnegie-Mellon, Stanford, MIT, and SRI--are still largely supported by ARPA. Smaller research programs have begun at some other universities such as Yale,

University of Illinois, University of Rochester, University of Texas, and University of Maryland. Besides conducting research, the universities play the important role of training the professionals who will be needed for AI applications.

RAND, University of Southern California's Information Sciences Institute, and Bolt, Beranek and Newman all engage in projects that are applications of AI ideas. Besides its basic research program, SRI is developing a strong applications arm. Several manufacturing companies are beginning projects that use AI technology in advanced industrial automation. Some of the aerospace companies have been tracking the field but have not yet participated actively.

Besides receiving funds from ARPA, AI research projects have been regularly supported by National Science Foundation, National Institutes of Health, National Aeronautics and Space Administration, Office of Naval Research, and Air Force Office of Scientific Research. ARPA has contributed perhaps somewhat over half of the total support.

The United States has been the major source of AI research and applications. The U.S. lead in computer technology plus support from ARPA has enabled AI research groups in the United States to dominate the field. Several strong foreign groups are emerging, however. The AI group at the University of Edinburgh is often regarded as on a par with the four major U.S. centers. Groups have begun in France (at Marseilles) and in Italy (at Pisa and at Milan). The Japanese have done some excellent AI work, notably at the government Electrotechnical Laboratories and at Hitachi and Mitsubishi. In addition, a recent international conference devoted to AI held in the Soviet Union (The Fourth International Joint Conference on Artificial Intelligence) revealed a rapidly emerging interest and capability in AI research at several Soviet laboratories.

The field has a journal called Artificial Intelligence in which many of the highest quality papers are published. In addition, biannual International Joint Conferences are held, and the proceedings

of these contain many of the important papers. Several collections of assorted papers and dissertations have been published, as well as a few textbooks devoted to theoretical foundations (Slagle, 1971; Nilsson, 1971; Jackson, 1974; Uhr, 1973; Hunt, 1975; Raphael, 1976)

c. Spin-Offs

AI research has produced a number of spin-offs in addition to its primary results. Since AI techniques are useful over such a broad range of applications, it often happens that a given application "disconnects" from its AI parentage and becomes a separate field. This type of spin-off has occurred in mathematical symbol manipulation and pattern recognition, to name two examples. Furthermore, AI research has often been synonymous with much of advanced computer science research. Thus, it was in AI research groups that list processing languages were developed, and an AI researcher (John McCarthy) worked out the basis of time-sharing.

2. Accomplishments

The purpose of this section is to describe briefly the major accomplishments of AI research. These have been of two types: (a) computer programs that demonstrate the usefulness of certain MI ideas in specific applications; and (b) conceptual milestones or tools that serve as key components of the technology.

a. Programs and Systems

Type (a) accomplishments include the following computer programs and systems:

* MACSYMA (Martin and Fateman, 1971)

  A system that assists applied mathematicians. It performs a wide range of tasks like symbolic integration and polynomial factoring, which are both tedious and very difficult to do correctly when the expressions become

large. MACSYMA evolved from a combination of earlier AI work in symbolic integration and from separate work in symbol manipulation. It has now been transferred to a consortium made up of Energy Resource Development Agency, National Aeronautics and Space Administration, National Institutes of Health, and the Navy Laboratories.

* DENDRAL (Buchanan et al., 1969; Buchanan and Lederberg, 1971)

  A system that infers chemical structure from organic mass spectrogram data. For some families of molecules, it operates more accurately and much more quickly than the best human mass spectrum analysts. An export version of DENDRAL, called EXODENDRAL, has been transferred to a national community of chemists who use it in an NIH-supported, Stanford-based computer resource for applications of AI to problems in biology and medicine.

* MYCIN (Shortliffe et al., 1973)

  A medical-assistant system that diagnoses bacterial infections and prescribes therapy. Its expertise is comparable to that of a general practitioner on these problems.

  Note: Both DENDRAL and MYCIN are examples of programs that incorporate and use "judgmental" or "intuitive" knowledge (in addition to traditional scientific information). An example of such a piece of knowledge used by MYCIN is:

  If:

  1) The gram stain of the organism is grampos, and
  2) The morphology of the organism is coccus, and
  3) The growth conformation of the organism is pairs

Then: There is suggestive evidence (.7) that the identity of the organism is streptococcus-pneumoniae.

Such systems can grow incrementally as additional knowledge is added.

* SHRDLU (Winograd, 1971)

  A program that could understand statements and answer queries typed in ordinary English into a terminal. Its domain of discourse concerned transporting simple geometric solids (called "blocks"). It could deal with a great variety of sentence constructions and became an "existence proof" that flexible language understanding systems could be built.

* LSNLIS (Woods et al., 1972)

  A system that can successfully answer typed unconstrained English questions about the properties of moon rocks returned by Apollo missions.

* SOPHIE (Brown et al., 1974)

  A system that makes effective use of problem solving and natural language understanding techniques to teach a technician to troubleshoot regulated power supplies.

* Speech-Understanding Systems, (Lesser et al., 1974; Woods, 1974; Walker, 1974)

  These systems are currently under development at Carnegie-Mellon; Bolt, Beranek and Newman; and SRI-SDC. Early versions have already been demonstrated, and by the end of 1976, they should meet most of the performance goals originally set for them (Newell et al., 1971).

* Perception of Three-Dimensional Solids (Roberts, 1963)

  An early program that processed digitized photographs of polyhedral objects to yield a composition of three-dimensional models that explained the scene. This demonstration of machine perception became an "existence proof" that image understanding systems be built.

* SHAKEY (Hart et al., 1972)

  A computer-controlled mobile robot with a TV camera that could navigate through doorways and around obstacles from one room into others. It used in automatic plan-generating system, called STRIPS (Fikes and Nilsson, 1971; Fikes et al., 1972), and had rudimentary abilities to store plans to use as components of more complex plans later.

* NOAH (Sacerdoti, 1975)

  An expert planning system that could generate and monitor the execution of complex, hierarchical plans. NOAH worked in the domain of electromechan-ical equipment and was used in a project to assist an apprentice technician in the repair of an air compressor.

* COPY (Winston, 1972)

  A combined vision and mechanical arm control program that could "look at" a complex tower of blocks, form an internal symbolic description of the structure, and then build a mirror image of the same structure.

* WAVE (Bolles and Paul, 1973)

  A program that used visual and tactile sensing in the automatic assembly of an automobile water pump.

* Industrial Automation Systems (Rosen et al., 1976; Finkel et al., 1975; Nevins et al., 1975)

  These automation systems, being developed under the sponsorship of National Science Foundation Research Applied to National Needs Program, have already attracted industrial attention. An example capability is illustrated by the SRI system that can visually identify randomly oriented castings coming down a conveyor belt. The system picks them up, packs the acceptable ones, and discards the ones with defects.

b. Concepts and Tools

In explaining the major conceptual successes of AI, it will be helpful to break the field down into manageable subparts. There are several ways in which this breakdown could be made. One method would be to divide the field according to the kinds of applications being pursued. From the point of view of a sponsor of AI research, such as ARPA, application categories could include command and control, electronic warfare, photointerpretation, software production, RPVs, and the like. Another method would be to use the categories that AI workers themselves use to divide themselves into interest groups. These are the major technical areas such as Expert Systems, Automatic Programming, General Reasoning, Data Management, Vision, and Natural Language. Each of these technical areas might be involved in several of the applications. Descending another level, there are a range of core topics that form the conceptual basis for AI.

The various example applications, technical areas, and core topics are tabulated in the chart below. In general, each application draws upon several of the technical areas, and each technical area draws upon several of the core topics. The following discussion of the conceptual successes of AI research will be in terms of the core topics listed in the chart.

EXAMPLE APPLICATIONS	AI TECHNICAL AREAS	CORE AI TOPICS
Command and Control	Expert Systems	Representations
Military Intelligence Applications	Automatic Programming	Control Structures
Automatic Photointerpretation Systems	General Reasoning	Heuristic Search
Electronic Warfare Applications	Data Management	Planning
Remotely Piloted Vehicles	Computer Vision	Perception
Software Production	Natural Language	Deduction
Automated Management Support Systems	Manipulation	Induction
Automated Manufacturing		Learning
		MI Languages and Systems

Representations

How should knowledge be acquired and represented so that it can best be used by a computer system? The types of knowledge for which representations are sought include:

* General statements of fact such as "all mammals have four legs."
* Natural language sentences, paragraphs, and stories.
* Effects of actions.
* Judgmental and uncertain statements such as "falling barometric pressure usually precedes rain."

Accomplishments in the development of representational concepts include:

* Demonstration that a set of assertional statements in predicate logic is a sufficient (if not always the most efficient) method of knowledge representation for many tasks. (Green, 1969.)
* Development of "semantic networks" of various types for representing concepts and their relationships. (Quillian, 1968; Simmons, 1973; Hendrix, 1975.)
* Demonstration of ways to represent knowledge as a procedure (i.e., a program). When the knowledge is needed, the procedure is run. This technique has been called "procedural embedding" of knowledge. (Hewitt, 1971; Winograd, 1971.)
* Development of "procedural networks" to represent plans of action. (Sacerdoti, 1975.)
* Demonstration that a large number of English verbs (including verbs of action, belief, and thought) can be represented by a much smaller number of conceptual entities appropriately modified by special case information in order

to capture the exact shade of meaning of the original verb. (Schank, 1973.)

* Demonstration that sets of rule-like quanta of knowledge form a sufficient basis for capturing the experienced judgments of experts about several domains including medicine, chemistry, and electronic circuit theory. These quanta are usually in the form of "productions," a well-understood construct in computer science. Production representations also allow the orderly development of large programs by incremental additions to the knowledge base. (Shortliffe, 1973; Davis and King, 1976.)

Control Structures

How can the conventional method of simple hierarchical control of computer programs be extended to enable more flexible encoding and use of diverse knowledge sources in a computer system?

Accomplishments in the development of concepts for control structures include:

* Use of pseudoparallel control regimes to shift the focus of a program's activities dynamically to operations of greatest current relevance. (This control strategy is sometimes referred to as "heterarchical control.") (Hewitt, 1971; Rulifson et al., 1971.)

* Use of pseudoparallel control regimes to investigate alternative sequences of actions. This approach allows a staged breadth-first search strategy to be built into a programming language. (Hewitt, 1971; Rulifson, et al, 1971.)

* Use of "pattern-directed function invocation" to select a subroutine at run time rather than to name a particular subroutine in advance. (Hewitt, 1971; Rulifson et al., 1971.)

* Use of "demons," functions invoked by pattern when a particular datum is added to or deleted from the data base. Pseudoparallel control, pattern-directed function invocation, and demons are significant features of the new MI languages (see ahead). (Hewitt, 1971; Rulifson et al., 1971.)

* Use of "production systems," sets of pattern directed functions that operate by writing to a common memory. This style of programming makes all side effects of computation explicit, and several large MI programs have now been written using this methodology. (Newell, 1972; Davis and King, 1976.)

* Use of augmented transition networks to control the syntactic parsing of English sentences. (Bobrow and Fraser, 1969; Woods, 1970)

How should the effects of the "combinatorial explosion" of exhaustive search be lessened?

* Use of "evaluation" functions to order the tip nodes of a search tree. (Many early workers; see, for example, Doran and Michie, 1966.)

* Discovery of the "alpha-beta" method of pruning game-trees. (Newell, Shaw, and Simon, 1958; Samuel, 1959; analyzed in detail by Knuth and Moore, 1975.)

* Use of "plausible move generators" to limit the branching of search trees by excluding all but the most likely paths.

* Use of "means-ends" analysis to select milestone nodes and branches toward which search can be focussed. (Newell, Shaw, and Simon, 1960.)

* Use of constraint satisfaction methods to eliminate combinations of nodes ruled out by given problem constraints. (Fikes, 1970; Waltz, 1972; Barrow and Tenenbaum, 1976.)

* Development of a rigorous mathematical theory of search using evaluation functions. (Hart, Nilsson, Raphael, 1968.)

Planning

How should plans of actions to achieve given goals be generated and executed, replanning as necessary? (Note: This problem can be viewed broadly to include the problem of automatic generation of computer programs.)

* Demonstration that uniform proof procedures of predicate logic are sufficient (if inefficient) to generate plans of action. (Green, 1969.)

* Invention of context frames to deal with the problem of keeping track of the effects of planned actions. (Fikes and Nilsson, 1971; Hewitt, 1971; Rulifson et al., 1971.)

* The development of planning systems in which the status of situations is represented as sets of assertions, and the effects of actions are represented as procedures that add and delete assertions. (Fikes and Nilsson, 1971; Hewitt, 1971; Rulifson et al., 1971.)

* The development of hierarchical planning systems to allow the top-down generation of plans at various levels of detail. (Sacerdoti, 1974; Sacerdoti, 1975.)

* The development of planning systems that use the strategy of "debugging" incorrect plans. (Sussman, 1975.)

* The development of ways to represent plans so that the plans themselves can be manipulated. This type of

representation allows the monitoring of plan execution, and the rapid modification of plans to fit changed situations. (Fikes et al., 1972; Sacerdoti, 1975.)

Perception

How can a description of the external world be derived from complex sensory input?

* Development of a variety of feature extraction techniques for acoustic (Schafer and Rabiner, 1975) and visual data (Duda and Hart, 1973).

* Development of mathematics and procedures for matching three-dimensional polyhedral models to two-dimensional visual data. (Roberts, 1963.)

* Development of methods for sensing and representing arbitrary three-dimensional objects. (Agin and Binford, 1973.)

* Development of a theory for the perception of scenes composed of arbitrary arrangements of polyhedral objects (including the effects of illumination, shadows, image formation, and occlusion). (See articles in the book edited by Winston, 1975.)

* The development of methods for using multiple sources of knowledge and constraint satisfaction techniques to handle interpretation problems common to speech and image understanding. (Erman and Lesser, 1975; Tenenbaum and Barrow, 1976.)

Deduction

How should programs deduce facts that are implied by other explicitly represented facts but are not themselves explicitly represented? (Note: In its general form, this problem is the same as the problem of proving theorems in mathematics.)

* The use of uniform proof procedures and the invention of an "answer extraction" mechanism to deduce answers to queries. (Green, 1969.)

* The representation of implicational statements as programs and their use to "reason forwards" and "backwards" in making deductions. (Hewitt, 1971; Rulifson et al., 1971.)

* The realization that efficient deduction procedures require expert knowledge about the problem domain.

Induction

How should programs make valid hypotheses about general situations based on specific observations?

* Formulation of the "hypothesize and test" paradigm and its use in programs like DENDRAL. (Buchanan and Lederberg, 1971.)

* Development of techniques for hypothesizing a generative grammar that might have produced a given set of symbol strings. (Feldman et al., 1969.)

* Use of production rules and techniques for combining uncertain evidence in systems that diagnose a patient's disease, given his symptoms. (Shortliffe, 1973.)

* Development of methods for hypothesizing a computer program that could have produced a given trace sample. (Hewitt, 1972.)

Learning

How should programs be written so that they automatically become more effective as they are run?

* Development of parameter adjustment techniques and their successful use in pattern recognition (Nilsson, 1965) and in improvement of the performance of game playing programs (Samuel, 1959, 1967).

* The development of techniques for making increasingly accurate structural descriptions of the definitions of objects from well-chosen examples. (Winston, 1970.)

* The development of methods to generalize and represent plans of action for subsequent use. (Fikes, et al, 1972.)

AI Languages and Systems

How should important strategies, processing methods, and representations be incorporated into more powerful and useful programming languages?

* The invention of list processing languages such as IPL-V and LISP. (Newell and Shaw, 1957; Mc Carthy, 1960.)

* The development of on-line, interactive programming environments such as INTERLISP. (Teitelman, 1974.)

* The invention of advanced AI languages such as MICROPLANNER, QA4, CONNIVER, SAIL, and QLISP. (See Bobrow and Raphael, 1974, for a review.)

3. Status of Hardware

Many AI programs require large computers. A typical AI computer installation consists of a PDP-10 processor with 250K or so of core memory backed up by disc secondary storage. Such an installation may cost around one million dollars. It is germane to ask whether hardware costs will continue to fall to the point where sophisticated AI applications can be achieved using computer systems costing, say, $25,000 or less.

There are several points to be made in this connection:

(1) Even if hardware costs do not continue to fall, AI applications programs will not require as much computational power as AI research programs. The applications programs often do not need the built-in

flexibility for easy modification, and do not need to run in a programming environment as complex as INTERLISP, for example. Experience on the SRI NSF industrial automation project (Rosen, 1976) has shown that it is entirely feasible to code complex arm control programs and part identification vision programs to be run on minicomputers (PDP-11's).

(2) For some types of AI applications that might currently require a million dollar computer installation, cost effectiveness is obtained if the installation supports 50 or so time-shared users ($20,000 per user). This type of usage might be entirely appropriate, for example, in a large headquarters operation.

The above two points are "worst case" arguments. Actually, it is reasonable to expect hardware costs to fall for the following reasons:

(1) "Fourth generation" LSI technology has not yet been used in large AI research computers.

(2) Preliminary investigations at Carnegie-Mellon University are beginning to show that MI programs can be run on a combination of several parallel minicomputers. These experiments point the way to AI systems composed of inexpensive microprocessors.

(3) Standard AI algorithms involving (for example) search, parsing, image, and waveform analysis can probably be implemented in inexpensive LSI circuitry. More generally, alternative computer architectures may provide hardware that is much better matched to AI styles of programming, with corresponding increases in efficiency.

BIBLIOGRAPHY

Agin, G. J. and T. O. Binford (1973), "Computer Description of Curved Objects," pp. 629-640, Proc. Third International Joint Conference on Artificial Intelligence, Stanford, California.

Barrow, H. and J. M. Tenenbaum (1976), "MSYS: A System for Reasoning About Scenes," Technical Note 121, SRI Artificial Intelligence Center, Menlo Park, California.

Bobrow, D. G. and B. Raphael (1974), "New Programming Languages for AI Research," ACM Computing Surveys, Vol. 6, No. 3, pp. 155-174 (September).

Bolles, R. and R. Paul (1973), "The Use of Sensory Feedback in a Programmable Assembly System," Memo CS-396, Comp. Sci. Dept., Stanford Univ., Artificial Intelligence Lab. Memo AIM-220 (October).

Brown, J. S. et al. (1974), "SOPHIE: A Sophisticated Instructional Environment for Teaching Electronic Troubleshooting (an example of CAI in AI)," BBN Report No. 2790, Bolt, Beranek, and Newman, Inc., Cambridge, Massachusetts (March).

Buchanan, B. G., G. Sutherland, and E. Feigenbaum (1969), "Heuristic DENDRAL: A Program for Generating Explanatory Hypotheses in Organic Chemistry," Machine Intelligence, Vol. 4, B. Meltzer and D. Michie, eds., pp. 209-254 (American Elsevier Publishing Company, New York, N.Y.

Buchanan, B. G. and J. Lederberg (1971), "The Heuristic DENDRAL Program for Explaining Empirical Data," Proc. IFIP Congress, Vol. 71, Ljubljana, Yugoslavia. Also Stanford Univ. AIM 141.

Davis, R. and J. King (1976), "An Overview of Production Systems," in Machine Representations of Knowledge, Proc. NATO Advanced Study Institute (D. Reidel Publishing Co.,).

Doran, J. and D. Michie (1966), "Experiments with the Graph Traverser Program," Proc. Roy. Soc. A, Vol. 294, pp. 235-259.

Duda, R. and P. Hart (1973), Pattern Classification and Scene Analysis (John Wiley & Sons, New York, N.Y.).

Erman, L. D., and V. R. Lesser (1975), "A Multilevel Organization for Problem Solving Using Many, Diverse, Cooperating Sources of Knowledge," pp. 483-498, Adv. Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR.

Feigenbaum, E. and J. Feldman, eds. (1963), Computers and Thought, (McGraw-Hill Book Company, New York, N.Y.)

Feldman, J. A. et al. (1969), "Grammatical Complexity and Inference," CS 125, Stanford Univ. Computer Science Dept.

Fikes, R. E. (1970), "REF-ARF: A System for Solving Problems Stated as Procedures," Artificial Intelligence, Vol. 1, No. 1.

Fikes, R. E. and N. J. Nilsson (1971), "STRIPS: A New Approach to the Application of Theorem Proving in Problem Solving," Artificial Intelligence, Vol. 2, pp. 189-208.

Fikes, R. E., P. E. Hart, and N. J. Nilsson (1972), "Learning and Executing Generalized Robot Plans," Artificial Intelligence, Vol. 3, pp. 251-288.

Finkel, R. et al. (1975), "An Overview of AL, A Programming System for Automation," pp. 758-765, Advance Papers of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR.

Green, C. (1969), "The Application of Theorem-Proving to Question-Answering Systems," Doctoral dissertation, Elec. Eng. Dept., Stanford Univ., Stanford, California (June). (Also printed as Stanford Artificial Intelligence Project Memo AI-96, June 1969.)

Hart, P. E. et al. (1972), "Artificial Intelligence--Research and Applications," Annual Tech. Report to ARPA, Contract DAHC04-72-C-0008, Stanford Research Institute, Menlo Park, California (December).

Hart, P., N. Nilsson, and B. Raphael, (1968), "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Sys. Sci. Cybernetics, Vol. 4, No. 2, pp. 100-107.

Hendrix, G. (1975), "Expanding the Utility of Semantic Networks through Partitioning," pp. 115-121, Adv. Papers 4th International Conference on Artificial Intelligence, Tbilisi, USSR (September).

Hewitt, C. (1971), "Procedural Embedding of Knowledge in PLANNER." Proc. 2d Intl. Joint Conf. on Artificial Intelligence, pp. 167-182, British Computer Society, London, England.

Hewitt, C. (1972), "Description and Theoretical Analysis (Using Schemata) of PLANNER: A Language for Proving Theorems and Manipulating Models in a Robot," Ph.D. thesis, Dept. of Math., MIT, Cambridge, Massachusetts.

Hunt, E. B. (1975), "Artificial Intelligence" (Academic Press, New York, N. Y.)

Jackson, P. C. (1974), "Introduction to Artificial Intelligence," (Petrocelli Books, New York, N.Y.)

Knuth, D. E. and R. W. Moore (1975), "An Analysis of Alpha-Beta Pruning," Artificial Intelligence, Vol. 6, No. 4, pp. 293-326 (Winter).

Kulikowski, C. A. and S. Weiss (1972), "The Medical Consultant Program--Glaucoma," Tech. Report TR-5, Computers in Biomedicine, Dept. of Comp. Sci., Rutgers University, New Brunswick, N.J. (July).

Lesser, V. et al. (1974), "Organization of the HEARSAY II Speech Understanding System," pp. 11-21, IEEE Symposium on Speech Recognition, Computer Science Dept., Carnegie Mellon University, (April).

Martin, W. A. and R. J. Fateman (1971), "The MACSYMA System," pp. 23-25, Proc. ACM 2d Symposium on Symbolic and Algebraic Manipulation, S. R. Petrick, ed., Los Angeles, California (March).

McCarthy, J. (1960), "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part 1," Comm. ACM, Vol. 3, No. 4, pp. 184-195 (April).

Moses, J. (1967), "Symbolic Integration," MAC Tech. Report 47, Project MAC, MIT, Cambridge, Massachusetts (December).

Nevins, J. et al. (1975), "Exploratory Research in Industrial Modular Assembly", Reports 1, 2, & 3, National Science Foundation Grant No. 6I39432X and ATA 74-18173 AOI (1 June 1973 to 31 August 1975).

Newell, A. (1972), "Production Systems: Models of Control Structures," Visual Information Processing, Wm. Chase, ed., (Academic Press, New York, N.Y.)

Newell, A. et al. (1971), "Speech Understanding Systems," Advanced Research Projects Agency Report, 1971. (Also published by North Holland Publishing Company, Amsterdam, The Netherlands, 1973.)

Newell, A. and J. C. Shaw (1957), "Programming the Logic Theory Machine," Proc. West. Joint Comp. Cont., pp. 230-240.

Newell, A., J. Shaw, and H. Simon (1958), "Chess Playing Programs and the Problem of Complexity," IBM J. Res. Develop., Vol. 2, pp. 320-335 (October).(Reprinted in Computers and Thought, Feigenbaum and Feldman, eds., pp. 39-70, McGraw-Hill Book Company, New York, N.Y., 1963.)

Newell, A., J. C. Shaw, and H. A. Simon (1960), "Report on a General Problem-Solving Program for a Computer," pp. 256-264, Information Processing: Proc. Intl. Conf. Information Processing, UNESCO, Paris. (Also printed in Computers and Automation, July 1959).

Nilsson, N. J. (1965), Learning Machines: Foundations of Trainable Pattern--Classifying Systems (McGraw-Hill Book Co., New York, N.Y.)

Nilsson, N. J. (1971), Problem-Solving Methods in Artificial Intelligence (McGraw-Hill Book Company, New York, N.Y.)

Quillian, M. R. (1968), "Semantic Memory," Semantic Information Processing, MIT Press, Cambridge, Massachusetts.

Raphael, B. (1976), The Thinking Computer (W. H. Freeman and Co., San Francisco, California).

Roberts, D. G. (1963), "Machine Perception of Three-Dimensional Solids," Tech. Report No. 315, MIT Lincoln Lab., Lexington, Massachusetts (May). (Also in Optical and Electro-Optical Information Processing, I. T. Tippett et al., eds., MIT Press, Cambridge, MA, 1965.

Rosen, C. A., et al. (1976), "Exploratory Research in Advanced Automation," Stanford Research Institute Fifth Report to the National Science Foundation (Grant GI-38100X1) (January).

Rulifson, J. F., R. J. Waldinger, and J. A. Derksen (1971), "A Language for Writing Problem-Solving Programs," Proc. IFIP, TA-2, pp. 111-115.

Sacerdoti, E. D. (1974), "Planning in A Hierarchy of Abstractions Spaces," Artificial Intelligence, Vol. 5, No. 2, pp. 115-135 (Summer).

Sacerdoti, E. (1975), "A Structure for Plans and Behavior," SRI Artificial Intelligence Center, Tech. Note 109 (August).

Samuel, A. (1959), "Some Studies in Machine Learning Using the Game of CHECKERS," IBM J. Research Devel., Vol. 3, pp. 211-229. (Reprinted in Computers and Thought, E. Feigenbaum and J. Feldman, eds., pp. 71-105, McGraw-Hill Book Company, New York, N.Y., 1963).

Schafer, R. W. and L. R. Rabiner (1975), "Parametric Representations of Speech," pp. 99-150, in Speech Recognition, D. R. Reddy (ed.). (Academic Press, New York, N.Y.)

Schank, R. C. (1973), "Identification of Conceptualizations Underlying Natural Language," pp. 187-247, in R. C. Schank and K. M. Colby (eds.), Computer Models of Thought and Language (W. H. Freeman and Co., San Francisco, California).

Shortliffe, E. H. et al. (1973), "An Artificial Intelligence Program to Advise Physicians Regarding Antimicrobial Therapy," Computers and Biomedical Research, Vol. 6, pp. 544-560.

Simmons, R. F. (1973), "Semantic Networks: Their Computation and Use for Understanding English Sentences," pp. 63-113, in R. C. Schank and K. M. Colby (eds.), Computer Models of Thought and Language (W. H. Freeman and Co., San Francisco, California).

Slagle, J. R. (1971), Artificial Intelligence: The Heuristic Programming Approach (McGraw-Hill Book Co., New York, N. Y.)

Sridharan, N. S. (1971), "An Application of Artificial Intelligence to Organic Chemical Synthesis," thesis, State Universiy of New York, Stony Brook, New York (July).

Sussman, G. J. (1975), "A Computer Model of Skill Acquisition," (American Elsevier Publishing Co., Inc., New York, N.Y.).

Teitelman, W. (1974), INTERLISP Reference Manual, Xerox and Bolt, Beranek and Newman. Copies available from Xerox Palo Alto Research Center, Palo Alto, CA.

Uhr, L. (1973), "Pattern Recognition, Learning and Thought," (Prentice-Hall, Inc., Englewood Cliffs, New Jersey).

Walker, D. E. (1974), "The SRI Speech Understanding System," pp. 32-37, IEEE Symposium on Speech Recognition, Computer Science Dept., Carnegie-Mellon University (April).

Waltz, D. C. (1972), "Generating Semantic Descriptions from Drawings of Scenes with Shadows," AI TR-271, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts (August).

Wegner, P., and W. Wulf, eds, The Impact of Research on Software Technology, in press.

Winograd, T. (1971), "Procedures as Representation for Data in a Computer Program for Understanding Natural Language," Tech. Report AI TR-17, MIT, Cambridge, Massachusetts (Published as Understanding Natural Language, Academic Press, New York, N.Y., 1972.)

Winston, P. H. (1970), "Learning Structural Descriptions from Examples," Tech. Report AI TR-231, Artificial Intelligence Laboratory, MIT, Cambridge, Massachusetts.

Winston, P. H. (1972), "The MIT Robot," Machine Intelligence, Vol. 7, pp. 431-463, B. Meltzer and D. Michie, eds., (American Elsevier Publishing Company, New York, N.Y.

Winston, P. H. (ed.), (1975), "The Psychology of Computer Vision," (McGraw-Hill Book Co., New York, N.Y.)

Woods, W. A. (1970), "Transition Network Grammars for Natural Language Analysis," Comm. ACM, Vol. 13, pp. 591-606.

Woods, W. A., R. M. Kaplan, and G. Nash-Webber (1972), "The Lunar Science Natural Language Information System: Final Report," BBN Report No. 2378, Bolt, Beranek and Newman, Inc., Cambridge, Massachusetts (June).

Woods, W. A. (1974), "Motivation and Overview of BBN SPEECHLIS: An Experimental Prototype for Speech Understanding Research," IEEE Symposium on Speech Recognition, Carnegie-Mellon University (April).

B. A Project Plan for a Computer Based Consultant System for Military Vehicle Maintenance*

by Nils J. Nilsson

1. Introduction

----

* This plan was developed under ARPA support for possible initiation in 1975. Although it was not funded and therefore not initiated as scheduled, it remains available for possible future revision and use.

a. Development of a Plan

The main goal of the Computer Based Consultant (CBC) project is to create the technology needed to develop expert systems in a variety of applications areas. Our strategy to achieve that goal is to select a specific applications area that is important in its own right and then to develop a demonstration system that performs well in that area. We have selected the task area of military vehicle maintenance.

The demonstration system developed under the CBC project will be an advanced experimental system. It will not be "human-engineered" or "ruggedized" to meet realistic military maintenance situations. Some of the transactions in the demonstration might not occur in real time. We will be attempting to integrate into the system a greater number of specific abilities than might be required in any particular application so that the resulting technology will span a variety of applications. Thus, our demonstration system will serve to illustrate the technology that has been developed and will be a springboard from which efforts can be launched to build prototypes for several specific applications.

The major subgoal of the project is to produce the demonstration system. A coordinate subgoal is to determine how this technology ought to be transferred to applications. Full documentation of all the results of the project plus the conclusions of the transfer study will be an important part of the whole program.

We have given a considerable amount of thought to defining appropriate target abilities for the demonstration system. These abilities can be grouped into four major areas: natural language (voice) communication; visual perception; planning and deduction; and troubleshooting. We have analyzed tape-recorded dialogs between human expert consultants and apprentice maintenance technicians to determine the requirements for these abilities. From these dialogs we have made our best guesses about which abilities could actually be incorporated

into computer programs over the next few years by a determined research effort and by incorporation of the results of other ongoing research in artificial intelligence.

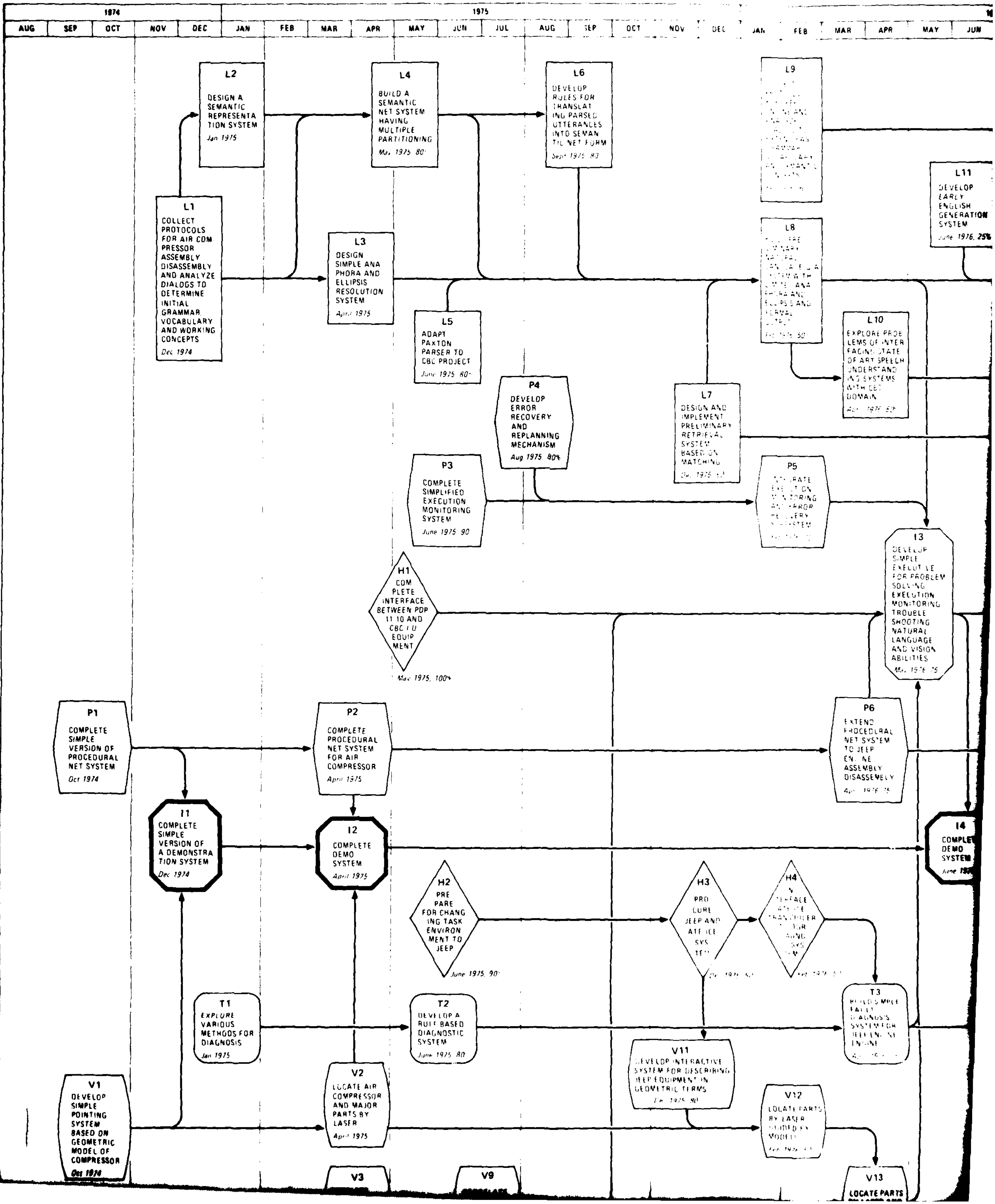
This analysis has resulted in estimated specifications for a demonstration system that we think can be achieved by June 1978. A full description of this system is given in Subsection 2. Achieving this system requires that we achieve each of a whole structure of subgoals as set forth in a rather thoroughly planned and scheduled set of tasks. The major purpose of this section is to describe this task structure in detail and to define each of the tasks as precisely as we can.

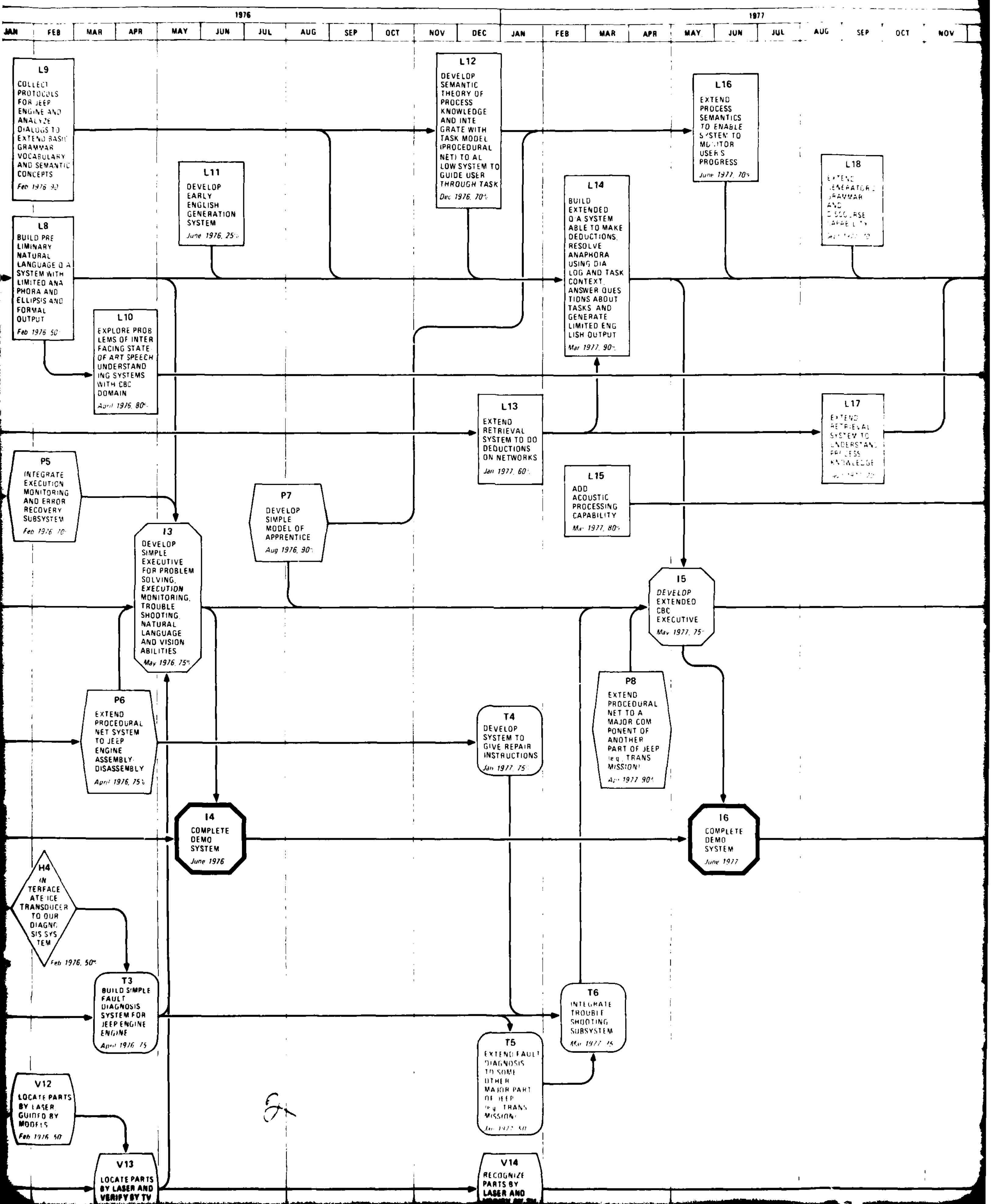
b. The Plan Structure

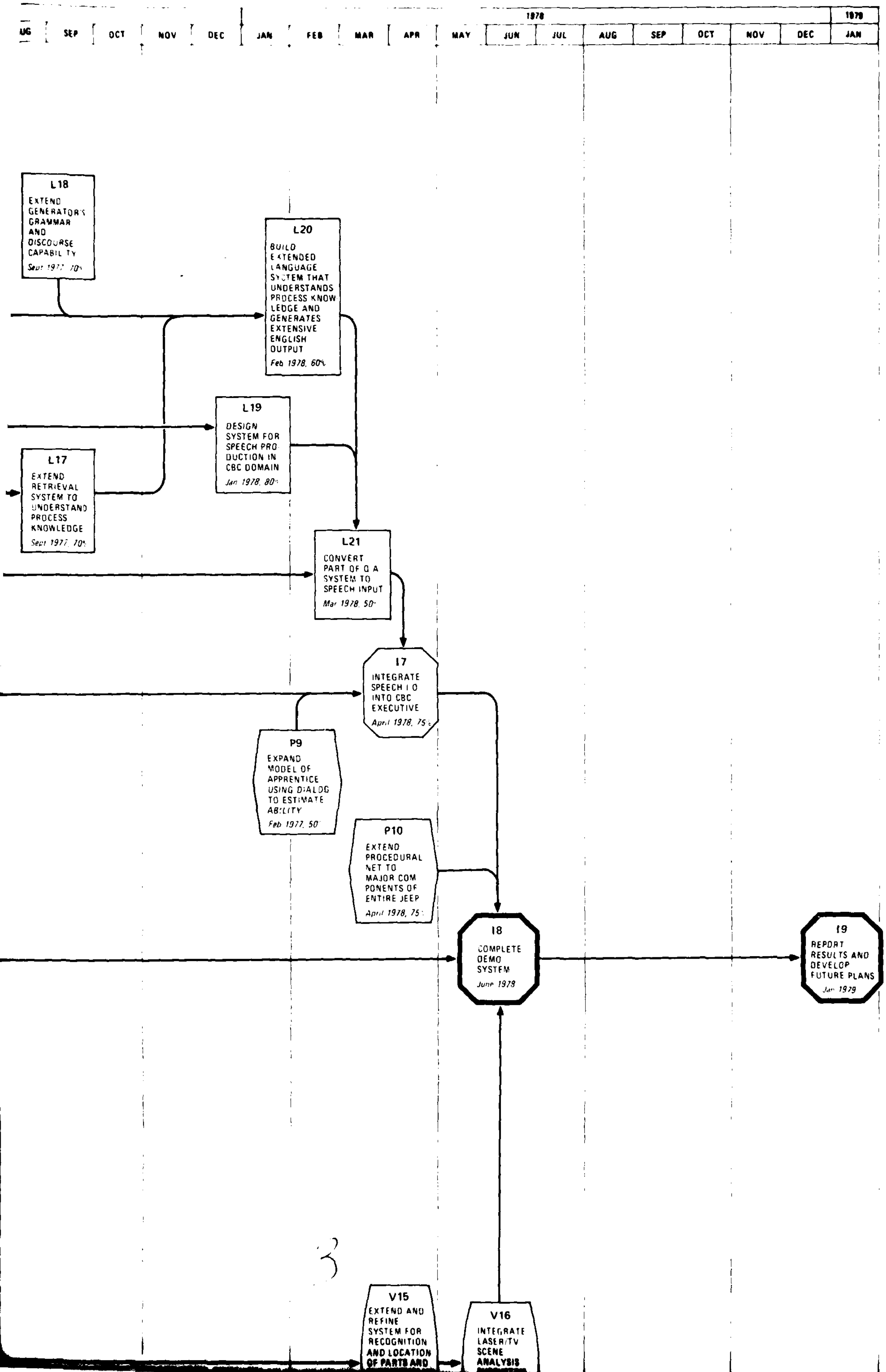
A complete view of our plan is shown in Figure 1. We represent the plan as a time network of tasks with arrows showing how the accomplishment of one task contributes toward another. For the purpose of illustrating continuity with previous work, we show some of the already completed tasks in the chart. At first glance, the network may appear rather complex, but some overview comments about it will help in understanding it.

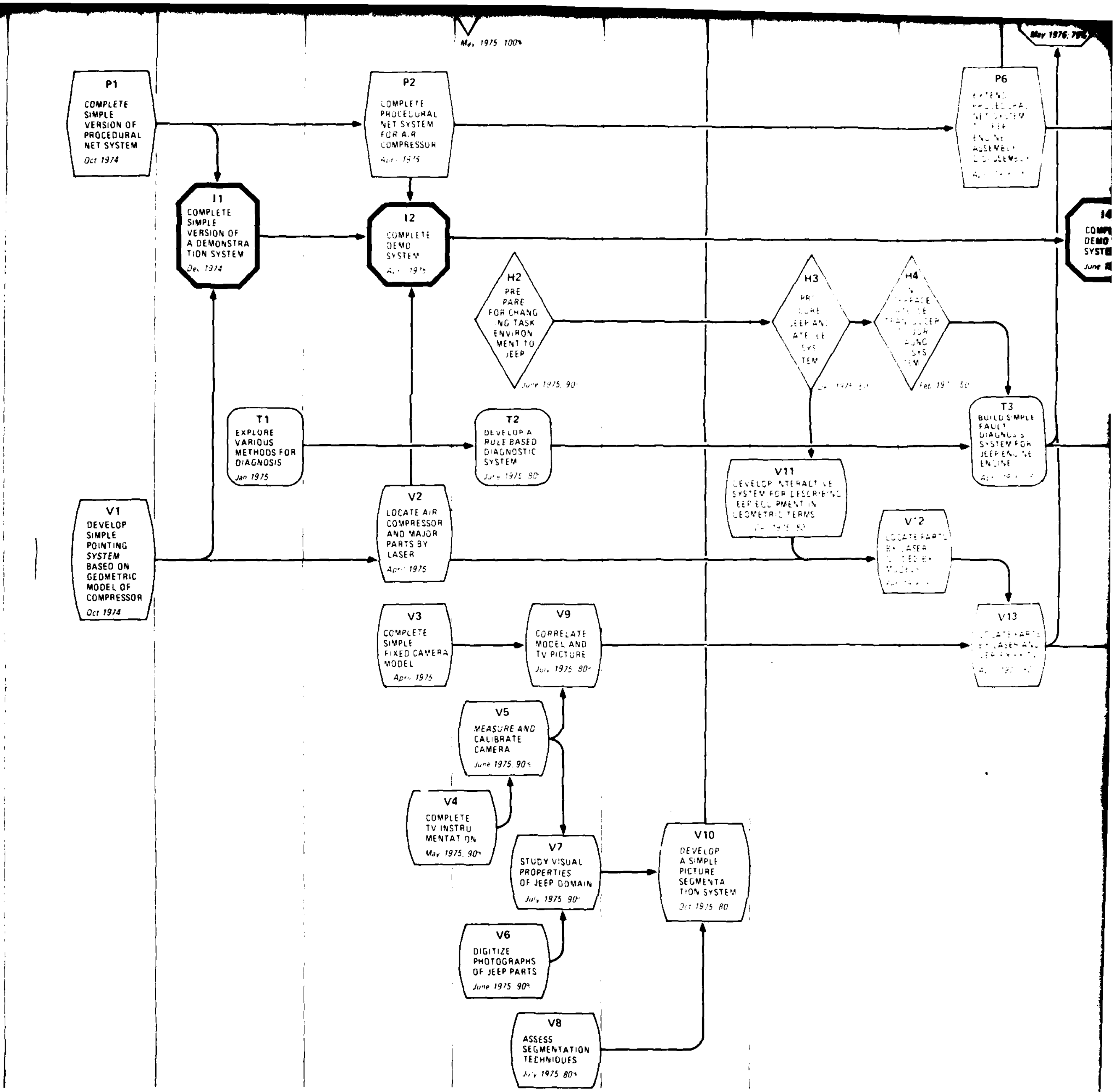
First, we would draw attention to a set of central tasks devoted to developing a series of progressively more complex demonstration systems ending with the June 1978 demonstration system. The tasks of completing these demonstration systems are indicated by heavy-outline boxes in the network of Figure 1. Each of the demonstrations is described in detail in Subsection 2 below.

We have divided the tasks into six groups: system integration; natural language communication; visual perception; problem solving and deduction; troubleshooting; and hardware interfacing. In the network of Figure 1, we indicate task group membership by differently shaped task boxes as explained by the key.

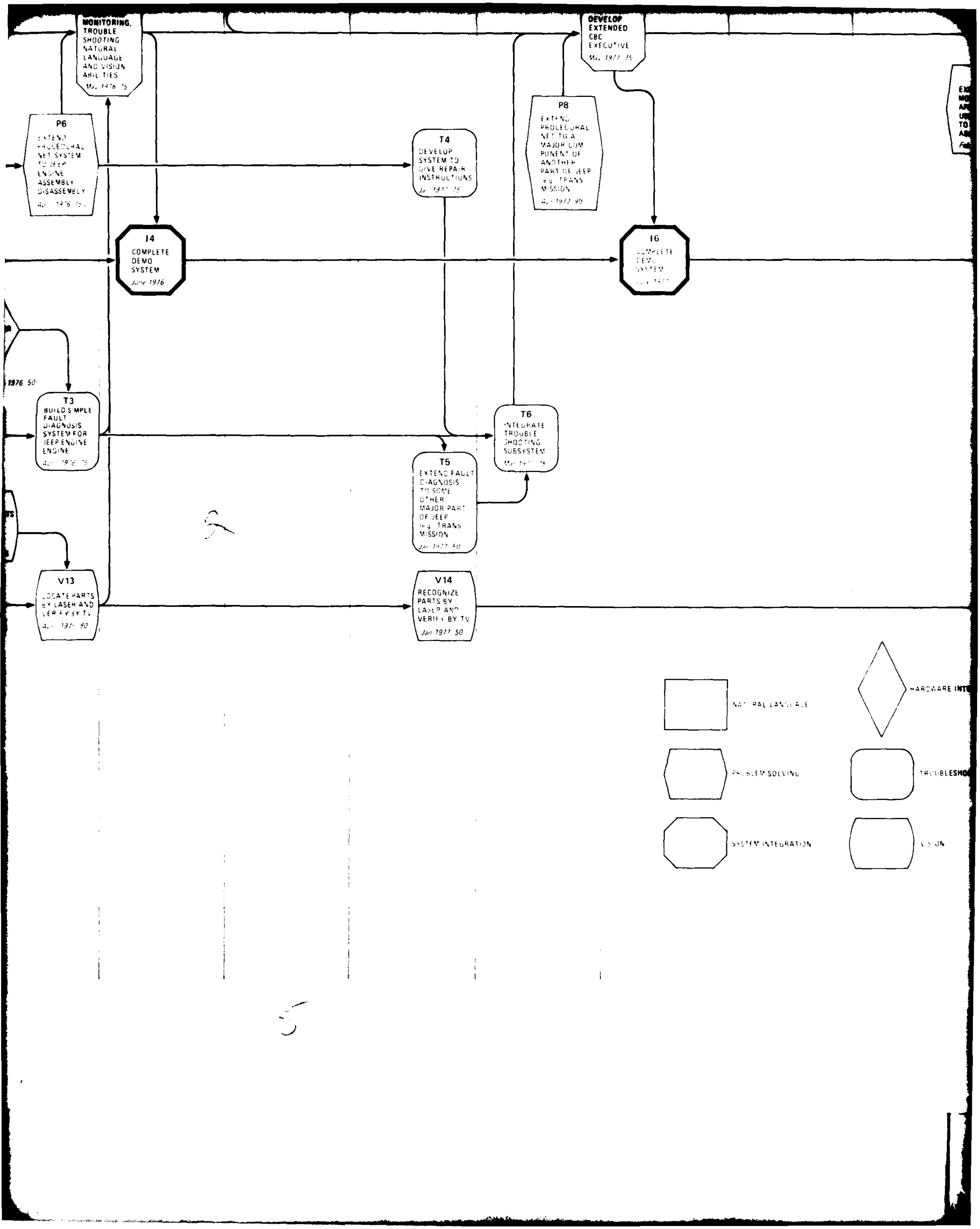

1974
AUG
SEP
OCT
NOV
DEC
1975
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC
JAN
FEB
MAR
APR
MAY
JUN
L2 DESIGN A SEMANTIC REPRESENTATION SYSTEM Jan 1975
L4 BUILD A SEMANTIC NET SYSTEM HAVING MULTIPLE PARTITIONING May 1975 80%
L6 DEVELOP RULES FOR TRANSLATING PARSED UTTERANCES INTO SEMANTIC NET FORM Sept 1975 83
L9
L11 DEVELOP EARLY ENGLISH GENERATION SYSTEM June 1976, 25%
L1 COLLECT PROTOCOLS FOR AIR COMPRESSOR ASSEMBLY DISASSEMBLY AND ANALYZE DIALOGS TO DETERMINE INITIAL GRAMMAR VOCABULARY AND WORKING CONCEPTS Dec 1974
L3 DESIGN SIMPLE ANAPHORA AND ELLIPSIS RESOLUTION SYSTEM April 1975
L8
L5 ADAPT PAXTON PARSER TO CBC PROJECT June 1975 80%
L10 EXPLORE PROBLEMS OF INTERFACING STATE OF ART SPEECH UNDERSTANDING SYSTEMS WITH CBC DOMAIN
P4 DEVELOP ERROR RECOVERY AND REPLANNING MECHANISM Aug 1975 80%
L7 DESIGN AND IMPLEMENT PRELIMINARY RETRIEVAL SYSTEM BASED ON MATCHING
P3 COMPLETE SIMPLIFIED EXECUTION MONITORING SYSTEM June 1975 90
P5
I3 DEVELOP SIMPLE EXECUTIVE FOR PROBLEM SOLVING EXECUTION MONITORING TROUBLE SHOOTING NATURAL LANGUAGE AND VISION ABILITIES
H1 COMPLETE INTERFACE BETWEEN PDP 11 10 AND CBC I/O EQUIPMENT May 1975, 100%
P1 COMPLETE SIMPLE VERSION OF PROCEDURAL NET SYSTEM Oct 1974
P2 COMPLETE PROCEDURAL NET SYSTEM FOR AIR COMPRESSOR April 1975
P6 EXTEND PROCEDURAL NET SYSTEM TO JEEP ENGINE ASSEMBLY DISASSEMBLY
I1 COMPLETE SIMPLE VERSION OF A DEMONSTRATION SYSTEM Dec 1974
I2 COMPLETE DEMO SYSTEM April 1975
I4 COMPLETE DEMO SYSTEM
H2 PREPARE FOR CHANGING TASK ENVIRONMENT TO JEEP June 1975, 90
H3 PROCURE JEEP AND ATE ICE SYSTEM
H4
T1 EXPLORE VARIOUS METHODS FOR DIAGNOSIS Jan 1975
T2 DEVELOP A RULE BASED DIAGNOSTIC SYSTEM June 1975 80
T3
V11 DEVELOP INTERACTIVE SYSTEM FOR DESCRIBING JEEP EQUIPMENT IN GEOMETRIC TERMS
V1 DEVELOP SIMPLE POINTING SYSTEM BASED ON GEOMETRIC MODEL OF COMPRESSOR Oct 1974
V2 LOCATE AIR COMPRESSOR AND MAJOR PARTS BY LASER April 1975
V12
V3
V9
V13 LOCATE PARTS


1976
1977
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC
JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV
L9
COLLECT PROTOCOLS FOR JEEP ENGINE AND ANALYZE DIALOGS TO EXTEND BASIC GRAMMAR VOCABULARY AND SEMANTIC CONCEPTS
Feb 1976 90
L8
BUILD PRELIMINARY NATURAL LANGUAGE Q A SYSTEM WITH LIMITED ANAPHORA AND ELLIPSIS AND FORMAL OUTPUT
Feb 1976 50
L10
EXPLORE PROBLEMS OF INTERFACING STATE OF ART SPEECH UNDERSTANDING SYSTEMS WITH CBC DOMAIN
April 1976, 80%
L11
DEVELOP EARLY ENGLISH GENERATION SYSTEM
June 1976, 25%
L12
DEVELOP SEMANTIC THEORY OF PROCESS KNOWLEDGE AND INTEGRATE WITH TASK MODEL (PROCEDURAL NET) TO ALLOW SYSTEM TO GUIDE USER THROUGH TASK
Dec 1976, 70%
L13
EXTEND RETRIEVAL SYSTEM TO DO DEDUCTIONS ON NETWORKS
Jan 1977, 60%
L14
BUILD EXTENDED Q A SYSTEM ABLE TO MAKE DEDUCTIONS, RESOLVE ANAPHORA USING DIALOG AND TASK CONTEXT, ANSWER QUESTIONS ABOUT TASKS AND GENERATE LIMITED ENGLISH OUTPUT
Mar 1977, 90%
L15
ADD ACOUSTIC PROCESSING CAPABILITY
Mar 1977, 80%
L16
EXTEND PROCESS SEMANTICS TO ENABLE SYSTEM TO MONITOR USER'S PROGRESS
June 1977, 70%
L17
EXTEND RETRIEVAL SYSTEM TO UNDERSTAND PROCESS KNOWLEDGE
L18
EXTEND GENERATOR'S GRAMMAR AND DISCOURSE CAPABILITY
P5
INTEGRATE EXECUTION MONITORING AND ERROR RECOVERY SUBSYSTEM
Feb 1976 70
P6
EXTEND PROCEDURAL NET SYSTEM TO JEEP ENGINE ASSEMBLY DISASSEMBLY
April 1976, 75%
P7
DEVELOP SIMPLE MODEL OF APPRENTICE
Aug 1976, 90%
P8
EXTEND PROCEDURAL NET TO A MAJOR COMPONENT OF ANOTHER PART OF JEEP (e.g., TRANSMISSION)
Apr 1977 90%
I3
DEVELOP SIMPLE EXECUTIVE FOR PROBLEM SOLVING, EXECUTION MONITORING, TROUBLE SHOOTING, NATURAL LANGUAGE AND VISION ABILITIES
May 1976, 75%
I4
COMPLETE DEMO SYSTEM
June 1976
I5
DEVELOP EXTENDED CBC EXECUTIVE
May 1977, 75%
I6
COMPLETE DEMO SYSTEM
June 1977
H4
INTERFACE ATE ICE TRANSDUCER TO OUR DIAGNOSIS SYSTEM
Feb 1976, 50%
T3
BUILD SIMPLE FAULT DIAGNOSIS SYSTEM FOR JEEP ENGINE
April 1976 75
T4
DEVELOP SYSTEM TO GIVE REPAIR INSTRUCTIONS
Jan 1977, 75%
T5
EXTEND FAULT DIAGNOSIS TO SOME OTHER MAJOR PART OF JEEP (e.g., TRANSMISSION)
Jan 1977 50
T6
INTEGRATE TROUBLE SHOOTING SUBSYSTEM
Mar 1977 75
V12
LOCATE PARTS BY LASER GUIDED BY MODELS
Feb 1976 50
V13
LOCATE PARTS BY LASER AND VERIFY BY TV
V14
RECOGNIZE PARTS BY LASER AND


1978
1979
AUG SEP OCT NOV DEC JAN FEB MAR APR MAY JUN JUL AUG SEP OCT NOV DEC JAN
L18
EXTEND GENERATOR'S GRAMMAR AND DISCOURSE CAPABILITY
Sept 1977, 70%
L20
BUILD EXTENDED LANGUAGE SYSTEM THAT UNDERSTANDS PROCESS KNOWLEDGE AND GENERATES EXTENSIVE ENGLISH OUTPUT
Feb 1978, 60%
L19
DESIGN SYSTEM FOR SPEECH PRODUCTION IN CBC DOMAIN
Jan 1978, 80%
L17
EXTEND RETRIEVAL SYSTEM TO UNDERSTAND PROCESS KNOWLEDGE
Sept 1977, 70%
L21
CONVERT PART OF Q A SYSTEM TO SPEECH INPUT
Mar 1978, 50%
I7
INTEGRATE SPEECH I O INTO CBC EXECUTIVE
April 1978, 75%
P9
EXPAND MODEL OF APPRENTICE USING DIALOG TO ESTIMATE ABILITY
Feb 1977, 50%
P10
EXTEND PROCEDURAL NET TO MAJOR COMPONENTS OF ENTIRE JEEP
April 1978, 75%
I8
COMPLETE DEMO SYSTEM
June 1978
I9
REPORT RESULTS AND DEVELOP FUTURE PLANS
Jan 1979
V15
EXTEND AND REFINE SYSTEM FOR RECOGNITION AND LOCATION OF PARTS AND
V16
INTEGRATE LASER/TV SCENE ANALYSIS
3



May 1975 100%
May 1976 70%
P1
COMPLETE SIMPLE VERSION OF PROCEDURAL NET SYSTEM
Oct 1974
P2
COMPLETE PROCEDURAL NET SYSTEM FOR AIR COMPRESSOR
P6
I1
COMPLETE SIMPLE VERSION OF A DEMONSTRATION SYSTEM
Dec 1974
I2
COMPLETE DEMO SYSTEM
H2
PREPARE FOR CHANGING TASK ENVIRONMENT TO JEEP
June 1975 90%
H3
H4
T1
EXPLORE VARIOUS METHODS FOR DIAGNOSIS
Jan 1975
T2
DEVELOP A RULE BASED DIAGNOSTIC SYSTEM
T3
BUILD SIMPLE FAULT
V1
DEVELOP SIMPLE POINTING SYSTEM BASED ON GEOMETRIC MODEL OF COMPRESSOR
Oct 1974
V2
LOCATE AIR COMPRESSOR AND MAJOR PARTS BY LASER
April 1975
V11
V12
V13
V3
COMPLETE SIMPLE FIXED CAMERA MODEL
April 1975
V9
CORRELATE MODEL AND TV PICTURE
V5
MEASURE AND CALIBRATE CAMERA
June 1975 90%
V4
COMPLETE TV INSTRUMENTATION
May 1975 90%
V7
STUDY VISUAL PROPERTIES OF JEEP DOMAIN
July 1975 90%
V10
DEVELOP A SIMPLE PICTURE SEGMENTATION SYSTEM
V6
DIGITIZE PHOTOGRAPHS OF JEEP PARTS
June 1975 90%
V8
ASSESS SEGMENTATION TECHNIQUES
July 1975 80%



MONITORING, TROUBLE SHOOTING NATURAL LANGUAGE AND VISION ABILITIES
DEVELOP EXTENDED CBC EXECUTIVE
P6
ASSEMBLY DISASSEMBLY
T4
DEVELOP SYSTEM TO GIVE REPAIR INSTRUCTIONS
P8
I4
COMPLETE DEMO SYSTEM
June 1976
I6
T3
BUILD SIMPLE FAULT DIAGNOSIS SYSTEM FOR JEEP ENGINE
T6
INTEGRATE TROUBLE SHOOTING SUBSYSTEM
T5
EXTEND FAULT DIAGNOSIS TO SOME OTHER MAJOR PART OF JEEP
Jan 1977 .50
V13
LOCATE PARTS BY LASER AND
V14
RECOGNIZE PARTS BY LASER AND VERIFY BY TV
Jan 1977 .50
SYSTEM INTEGRATION

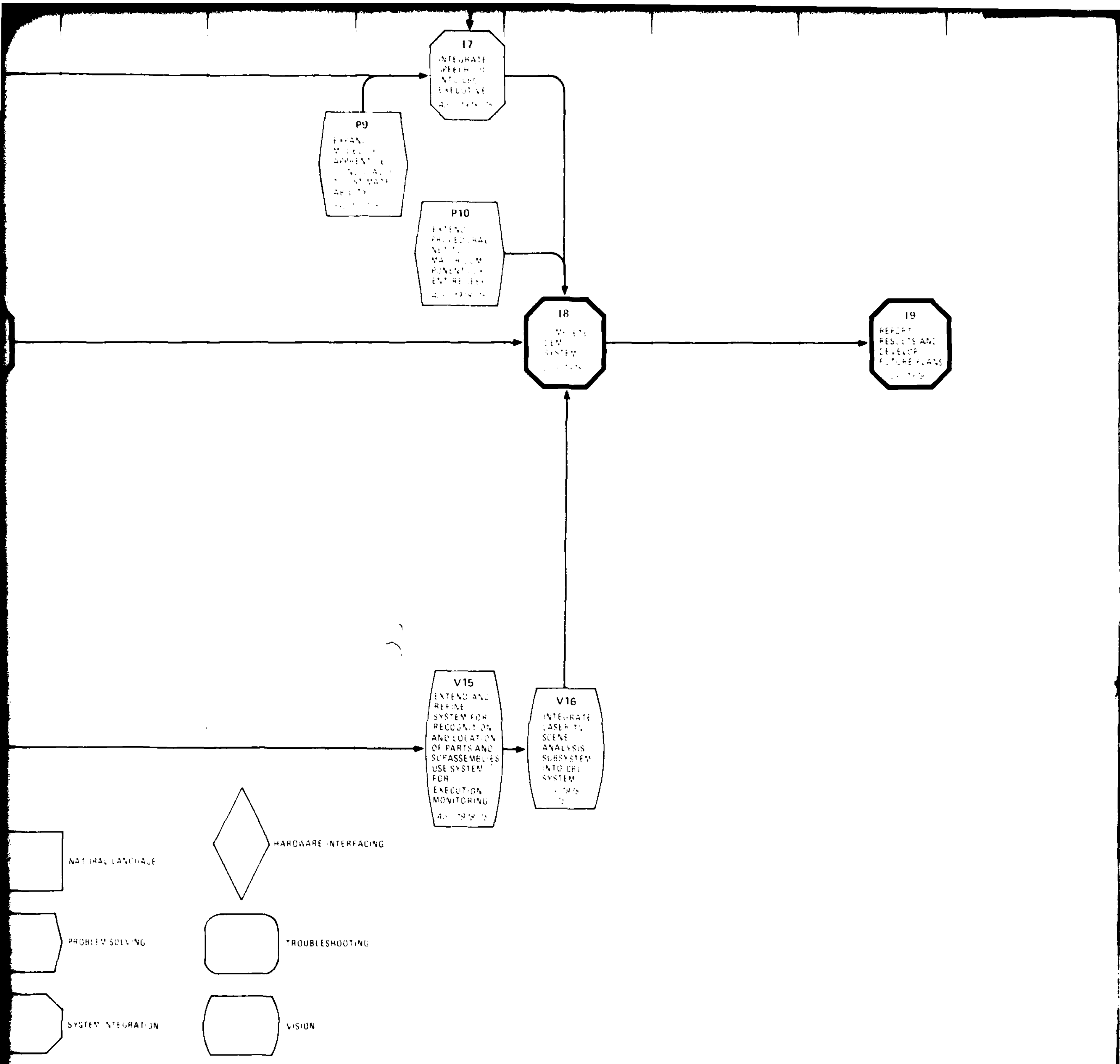


FIGURE 1 SCHEDULE FOR THE CBC PROJECT

A good way to read the network is to look backward from each of the demonstration systems and examine the subtasks involved in producing each demonstration. For the purpose of continuity, we include in the network some recently tasks.

Subsection 2 the demonstration systems, Subsection 3 gives a short description of each of the other tasks, and Subsection 4 we gives a brief review of the importance of equipment maintenance as a task area.

c. General Comments about the Plan

A few observations must be made if a research plan, such as the one presented here, is to be interpreted correctly. First, we emphasize that the plan is a plan for research. It is a plan to produce new technology rather than one to produce a system based on existing technology. Thus, its goals and subgoals must necessarily be rather tentative. To indicate this degree of uncertaintly, we have included a percentage probability of successful completion for each task shown in Figure 1. These probabilities are based on the assumption that each of the precursor tasks has been substantially accomplished.

The probabilities are such that we almost certainly will not arrive at our final goal along the precise paths indicated in Figure 1. Nevertheless, we are confident that we can arrive at a highly worthwhile demonstration system by June 1978. As we meet failures along some paths, we will replan, redefine, and take advantage of other opportunities (impossible to foresee now) along other paths. This process is common to any research effort. At any stage of the process, we will have an up-to-date plan--similar to the one in Figure 1--guiding the research.

2. Demonstration Systems

This section describes the general properties of the demonstration systems that form the key milestone steps of the project.

This description will be in summary form here; the demonstrations themselves are more completely, if only implicitly, specified by the schedule of component abilities discussed in the next section.

We will begin with a detailed description of the current (April 1975) system and then discuss the added abilities to be incorporated into each of the subsequent ones. (See Nilsson, N. J. et al., 1975 Annual Report to ARPA.) for a detailed description of our current status.)

a. The April 1975 Demonstration System

The present demonstration system communicates with an "apprentice" in a workstation containing the following items:

A workbench with a tool box and tools.

A round table with a turntable top on which is placed a small air compressor; this is located in the middle of the room with access from all sides.

A computer terminal.

A microphone headset with a long cord which will reach to any point in the room.

A speaker/amplifier.

A TV camera, mounted near the ceiling on a movable pan/tilt head.

A TV display (RAMTEK) upon which is displayed a TV picture of the air compressor, with a superimposed line model.

A laser/rangefinder mounted near the TV camera.

A lighted wand.

In describing a typical run of the demonstration, let us assume that the air compressor is in a partially disassembled condition. The belt housing cover and belt are removed and lying on the workbench. The pump bolts are removed, and the pump is turned away from its normal orientation.

The CBC system begins by asking the apprentice to "please assemble the air compressor." (This request is made through a VOTRAX speech synthesizer.) At any stage of the process, the apprentice may communicate vocally using any of the following phrases:

OK indicates the apprentice has performed the requested task or subtask.

HOW indicates more detailed instructions are needed.

WHY indicates a desire to know the motivation for the particular instruction just received.

HUH or WHAT or PLEASE-REPEAT requests that the last command be repeated.

WHERE IS THE ... or SHOW ME THE ... followed by the name of a component will result in positioning of the laser rangefinder beam on that component.

WHAT IS THIS, coupled with pointing the lighted wand at a component, will cause the CBC system to use its TV camera to look at the wand and then identify the component it points to.

BREAK, PAUSE, or WAIT will cause an interrupt in the program execution so that an experimenter can use the terminal to query the program about the state of things.

We are currently using the VIP/100 phrase recognition system to receive these inputs and a VOTRAX VS-6 phoneme generator to produce the computer's "voice."

The CBC contains a system for planning assembly or disassembly of the air compressor. The plan is represented by a structure called a procedural net. In this demonstration, it is built from the pre-specified disassembled state of the compressor described above. Each step of the assembly plan can be given at several different levels of detail, depending upon the needs of the particular apprentice. The program keeps an internal model of the "connectedness" of the different components of the air compressor. It also contains a graphical model of the shapes, relative positions, and locations of the different components. At present, these models are limited. The graphical model "knows about" the pump, motor, pulleys, pressure switch, belt housing frame, platform, tank, and table. The connectivity model "knows about" all the visible components, but the two models do not interact with each other.

The capabilities of the present system can probably be best illustrated by the following partial transcript of an actual dialog between the CBC and an apprentice:

E: TRY TO ACHIEVE ASSEMBLED AIR COMPRESSOR
A: HOW?
E: THE TOP LEVEL PLAN IS .. .. INSTALL PUMP .. INSTALL PUMP BRACE .. INSTALL PUMP PULLEY .. INSTALL BELT .. INSTALL BELT HOUSING COVER ..
E: READY FOR QUESTIONS
A: SHOW ME THE PUMP
(Laser beam moves to point at pump)
E: READY FOR QUESTIONS
A: WHAT IS THIS? (Pointing with wand at tank platform)
E: TANK PLATFORM
E: READY FOR QUESTIONS
A: NONE
E: INSTALL PUMP
A: HOW?
E: CONNECT PUMP TO PLATFORM
A: HOW?
E: POSITION PUMP ON PLATFORM
A: HOW?
E: PLACE THE PUMP ON THE PLATFORM IN SUCH A WAY THAT HOLES IN THE PUMP ARE MATCHED UP WITH HOLES IN THE PLATFORM
A: OK
E: ATTACH PUMP TO PLATFORM
A: HOW?
E: USE BOLTS TO ATTACH PUMP TO PLATFORM
A: HOW?
E: GET FOUR 1&1/4 INCH 5/16 HEXAGONAL BOLTS AND FOUR 9/16 FLAT WASHERS AND FOUR SIZE 5/16 HEXAGONAL NUTS
A: OK
E: LOOSELY FASTEN THE FOUR BOLTS
A: HOW?
E: START IN ONE BOLT
A: HOW?
E: PUT THE BOLT THROUGH A HOLE IN PUMP BASE, PLATFORM, WASHER, AND NUT, AND SCREW THE NUT CLOCKWISE ONTO THE BOLT AS FAR AS YOU CAN
A: OK
E: LOOSELY FASTEN THE OTHER THREE BOLTS
A: OK
E: TIGHTEN THE FOUR BOLTS
A: HOW?
E: TIGHTEN ONE BOLT
A: HOW?
E: FIND A SIZE 9/16 WRENCH TO FIT THE NUT
A: OK
E: FIND A SIZE 1/2 INCH WRENCH TO FIT THE BOLT HEAD
A: OK
E: USE THE FIRST WRENCH TO HOLD THE NUT FIRMLY
A: OK
E: USE THE SECOND WRENCH TO TURN THE BOLT CLOCKWISE UNTIL FAIRLY TIGHT

A: OK
E: TIGHTEN THE OTHER THREE BOLTS
A: OK
E: INSTALL PUMP BRACE
A: WHY?
E: NEXT STEPS ARE TO INSTALL PUMP PULLEY AND INSTALL BELT AND
INSTALL BELT HOUSING COVER
A: WHY?
E: IN ORDER TO TRY TO ACHIEVE ASSEMBLED AIR COMPRESSOR
A: OK
E: INSTALL PUMP BRACE
A: HOW?
E: CONNECT PUMP BRACE TO BELT HOUSING FRAME
A: OK
E: CONNECT PUMP BRACE TO PUMP
A: OK
E: INSTALL PUMP PULLEY
A: HOW?

.
.
.

E: ATTACH THE PUMP PULLEY TO THE SHAFT OF THE
PUMP
A: HOW?

.
.
.

E: INSERT THE SHORTER HEXAGONAL END OF THE
ALLEN WRENCH INTO THE HEXAGONAL HOLE OF THE
SETSCREW AND THEN TURN THE SETSCREW CLOCKWISE
UNTIL TIGHT
A: OK

.
.
.

E: INSTALL BELT
A: HOW?

.
.
.

E: IF YOU HAVE TROUBLE GETTING THE BELT AROUND
THE PUMP PULLEY, TRY MOVING THE MOTOR WHICH
MAY MEAN LOOSENING THE FASTENINGS TO ALLOW
IT TO SLIDE SIDEWAYS
A: OK
E: ATTACH BELT TO MOTOR PULLEY AND ATTACH BELT
TO PUMP PULLEY
A: HOW?
E: MAKE BELT VERY TIGHT BY ADJUSTING THE
POSITION OF THE MOTOR
A: OK

E: INSTALL BELT HOUSING COVER
A: OK
E: THANK-YOU
Finished

b. The June 1976 Demonstration System

The domain of expertise will shift from the air compressor to a component subsystem of a jeep. Also, several important abilities will be added to the system during the several months preceding June 1976. First there will be a rudimentary ability to communicate with the system, using natural language input. This input will be in text form and will consist of simple queries about the status of the equipment and about the properties of components and tools. For this system, the queries will be answered by performing reasonably straightforward matching operations on a semantic knowledge net. The answers themselves will be expressed in some formal fashion, i.e., not English.

The 1976 system will have some ability to monitor the execution of tasks given to the apprentice. We will use information provided by the apprentice himself as well as information obtained from the laser range finder. When these sources reveal that the apprentice failed to carry out a step correctly, the system will automatically replan what should be done to get the task back on the track. Revised instructions will then be given to the apprentice.

The planning system using the procedural net will be able to generate instructions for assembly and disassembly of parts of the jeep, most probably the engine.

The 1976 system will also be able to give advice about troubleshooting and maintenance. (It will not yet be giving advice about repair of faults.) The system will probably assume that there is at most one fault in the system to be diagnosed. Our plans are that the system should be able to diagnose several types of failures that prevent the jeep engine from starting.

The laser pointing system will be able to point out various parts of the jeep. The pointing system will be based on geometric models of the jeep and its parts. We will have more robust methods of calibrating the model, using television input and expanded ability to model additional jeep subsystems interactively.

All of these added abilities, together with the ones inherited from the 1975 system, will be integrated in a more flexible and efficient executive system that will permit extensive communication between the various subsystems.

c. The June 1977 Demonstration System

For this system we hope to extend the domain of expertise to include additional subsystems of the jeep, perhaps the transmission. The natural language abilities will be greatly augmented although all transactions will still be taking place in text. There will be an extended ability to handle ellipsis and anaphora, using the dialog history and the task context. The system will be able to answer questions about tasks, and will be able to generate limited English output. The deductive processes will be adequate to allow question-answering that requires simple inferences (in addition to matching).

The planning system will be extended to handle assembly and disassembly of more of the jeep. In addition, it will develop and use a simple model of each apprentice's ability to understand it.

The 1977 system will incorporate a troubleshooting system that will help the apprentice zero in on any of a number of faults in the jeep engine and possibly the transmission also. It will then suggest how the trouble should be repaired, and will use the planning system to generate precise instructions for doing so if needed.

The visual abilities of the system will permit the recognition of various parts of the jeep so that, for example, the apprentice will be able to hold up a connecting rod in front of the TV camera and request the system to identify it. We will be able to make

more extensive use of vision to help monitor the apprentice's progress in the task. For example, the visual system will be able to check to see if certain assembly steps have been performed.

d. The June 1978 Demonstration System

In the final system of this series, we hope to add a number of remaining features which, when taken together with the previous abilities, should result in a rather impressive demonstration. The main addition will be the ability to use a limited subset of spoken English. The text-based system will be converted in part (depending on manpower resources) to a speech-based system. The quality of speech understanding to be incorporated into the system will be roughly comparable to that of the 1976 Speech Understanding Systems projects. Our system will probably have a greater ability to deal with task-related questions, especially about processes.

The 1978 system should be able to give advice about repairing several different subsystems on the jeep and to answer questions that arise during the troubleshooting and repair processes. The planning system for assembly and disassembly will be able to guide the apprentice in a manner highly matched to his own ability level.

We hope by 1978 to have a rather thorough visual question-answering system--that is, a system that can use the TV/laser system to answer questions such as "Where is the wheel puller?"

3. Description of Task Areas

This section presents brief descriptions of each of the tasks shown in Figure 1. In addition, some of these tasks themselves have other subgoals either too detailed or too problematical to include in Figure 1. These will be described here and shown in additional figures. Each task has been given a number to help tie the figures and descriptions together.

a. System Integration

The tasks shown in Figure 1 are as follows:

I1 - Develop Demonstration System by December 1974.

(This was a preliminary version of the April '75 system.)

I2 - Develop Demonstration System by April 1975 (described in previous section).

I3 - Develop simple executive for problem solving, execution monitoring, troubleshooting, natural language, and vision abilities.

A LISP program will be written which will be, in a sense, a problem solver that will determine when to invoke different abilities of the consultant. This executive must allow for additions of new abilities as they become available, and for improvement in existing abilities of the consultant. Continuing effort will be required to keep the executive abreast of later developments. (See descriptions of tasks I5, I6.1, and I7.)

I4 - Develop Demonstration System by June 1976 (described in last section).

I5 - Develop Extended CBC Executive

Continuing the task of I-3 above, this task will incorporate the newly developed abilities to produce an improved executive.

I6 - Develop Demonstration System by June 1977 (described in last section).

I7 - Integrate Speech I/O Into CBC Executive

The consultant system developed through 1977 will continue to use simple devices and programs for speech understanding and spoken output. A parallel effort will be producing a much more sophisticated speech ability which will be placed under the control of the CBC Executive Program.

I8 - Develop Demonstration System by June 1978 (described in last section).

Figure 2 shows some subsidiary integration tasks. These are described as follows:

I4.1 - Transfer VIP to PDP 11-10

The VIP voice input device currently sends signals to the PDP-10 via a PDP-15 computer. At present, the VIP is loaded with messages and training data by the laborious

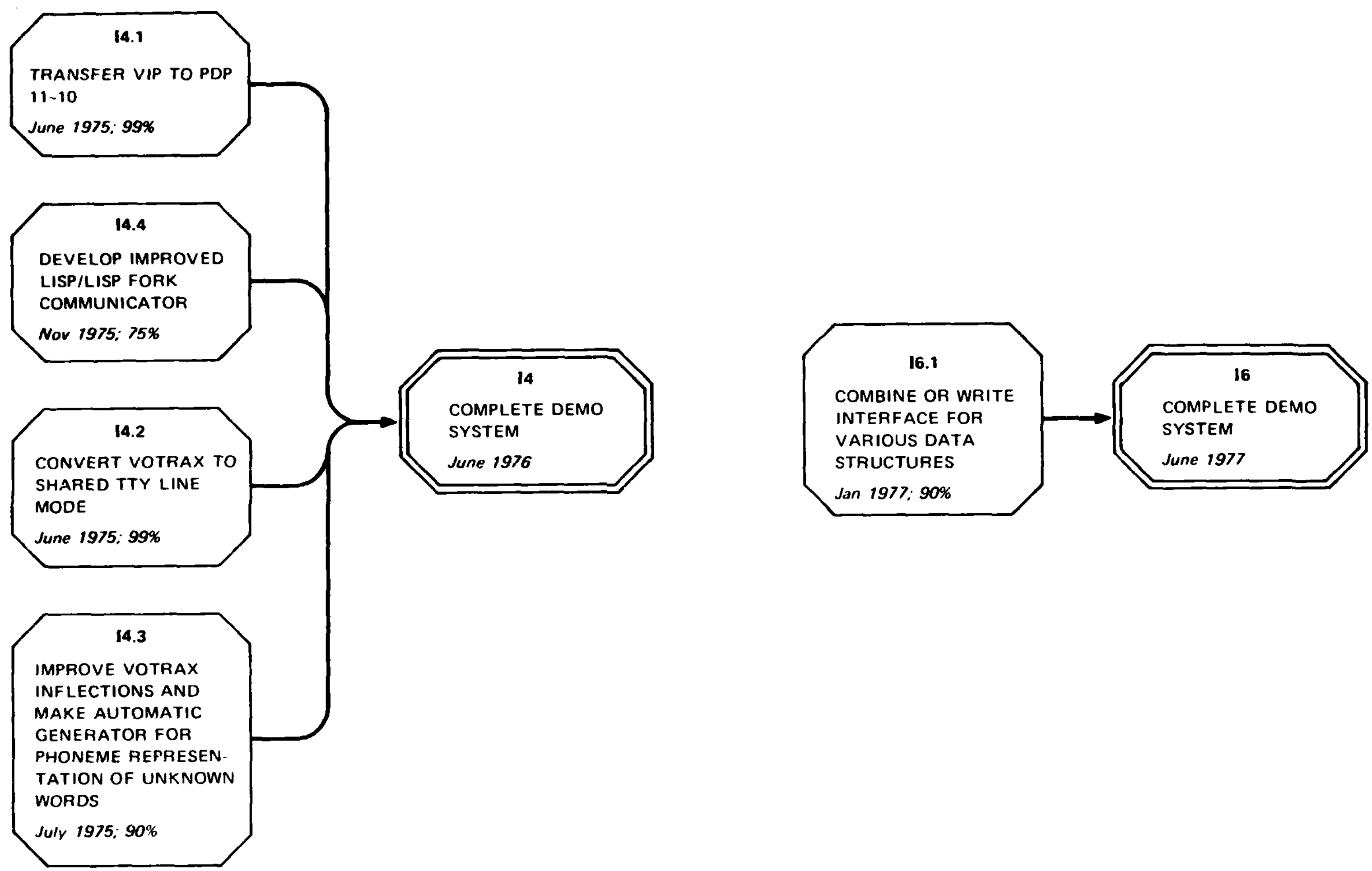


FIGURE 2 ADDITIONAL INTEGRATION TASKS

reading of punched paper tapes. The PDP 11-10 will use the ARPAnet to communicate with the PDP-10, thus eliminating the use of the PDP-15 and allowing the VIP's data to be stored on the file system of the PDP-10. This transfer should be a fairly straightforward, since some of the capabilities have already been developed by the SRI-AIC Industrial Automation project.

I4.2 - Convert VOTRAX to Shared TTY Line Mode

currently receives signals from a standard teletype line. However, it is possible, with a hardware modification available from the manufacturer, to allow the VOTRAX to share a line with a regular terminal. This shared mode requires some reprogramming in the way characters are sent across the line.

I4.3 - Improve VOTRAX Inflections and Develop Automatic Generator for Phoneme Representation of Unknown Words

The VOTRAX will likely be used for spoken output for several more years. A desirable goal is to improve its speech to make it more readily understood. A continuous sentence-based inflection scheme will be developed which will attempt to emulate, to a small degree, the typical sentence inflections. Another program, which will "guess" at the phoneme representation for an unknown word, will be written to eliminate the awkwardness that occurs when the CBC program encounters a new word.

I4.4 - Develop Improved LISP/LISP Ford Communicator

It is clear that the CBC system will soon grow much too big to be handled within a single INTERLISP program. However, because the many important components of the consultant will need to communicate, it will be necessary to have the facilities for LISP/LISP data transfer, no matter how inefficient. It is hoped that in the future a reasonably efficient, general ability to do this can be attained.

I6.1 - Combine or Write Interface for Various Representations

Each of the different components of the consultant will employ some sort of data structure that contains its "world model." Since the components must all interact, and since there is only one "real world," it will be necessary to have the ability to interface these data structures, both to allow ease in keeping them up to date and consistent, and to allow different kinds of reasoning about the problem domain. An important goal of the CBC project is not merely to learn to live with multiple representations, but also to take advantage of them.

We estimate that the total effort needed for the integration tasks (including the demonstrations) will require a rate of about one person per year augmented by some programming help.

b. Problem Solving and Deduction

The tasks shown in Figure 1 are as follows:

P3 - Complete Simplified Execution Monitoring Scheme

As the apprentice moves through the task, the system will maintain an updated world model of the current state of progress. The system or the apprentice may query this model. This system will ensure that steps that have been resequenced have been performed at the proper time.

P4 - Develop Error Recovery and Replanning Mechanisms

When the apprentice finds it impossible to perform a specified action, the system will assume that an error has occurred at some previous time, and that the actual state of affairs differs from its world model. The system will use the hierarchical structure of the procedural net to ask questions about the actions that the apprentice has performed. These questions may deal with greater levels of detail than the discussion that initiated the original execution. Appropriate algorithms will be developed for focusing in on the problem with a minimum of interaction.

When the error has been localized, the procedural net will be patched to enable the apprentice to get back on the track. This plan revision will be accomplished with a minimum of replanning, and a maximum use of existing plans.

P5 - Integrate Execution Monitoring and Error Recovery Subsystem

A single executive will be created to handle all the interactions between apprentice and the task model represented by the procedural net.

P6, P6.4 - Extend Procedural Net System to Jeep Engine Assembly/Dissassembly

SOUP code will be written to model the assembly and disassembly of the major components of the jeep engine.

P7 - Develop Simple Model of the Apprentice

A rudimentary model of the apprentice's level of competence at various subtasks will be developed. The model will be built up by noting his progress through the

procedural net. The model will be used to determine the level of detail to which the initial plans will be built. The model will be updated dynamically.

P8 - Extend Procedural Net to Another Jeep Component

We will develop the appropriate semantics, and then write and debug more SOUP code.

P9 - Expand Model of Apprentice Using Dialog to Estimate Ability

P10 - Extend Procedural Net to Major Components of Entire Jeep

We will develop more semantics, and write and debug still more SOUP code.

Figure 3 shows some subsidiary problem solving tasks. These are described as follows:

P6.1 - Develop Model of Freedom of Movement to Use in Computing Preconditions

We have developed a model of the pump assembly of the air compressor--a model that can be used to answer questions regarding the "freedom of movement" of individual parts in the subassembly. This model will be used, for example, to establish preconditions for removing parts.

P6.2 - Extend Assembly/Disassembly Semantics to Include Preconditions

We will include the preconditions for each action in the procedural semantics (the SOUP code) for assembly and disassembly actions. This effort may require some modifications to the critic algorithms of the planner. At this point, all the planning aspects will have been completed.

P6.3 - Collect Semantics for Jeep Engine Assembly

A hierarchy of relations will be developed for describing states of partial assembly of the components of the jeep engine. Actions will be specified for converting from one state to another.

P6.5 - Expand Ability to Answer Questions About Plans

We will use the procedural net to answer queries about the teleology of actions and subplans, and about the effects of hypothesized actions.

We estimate that the total effort needed for the problem solving and deduction tasks will requiire a rate of about two persons per year.

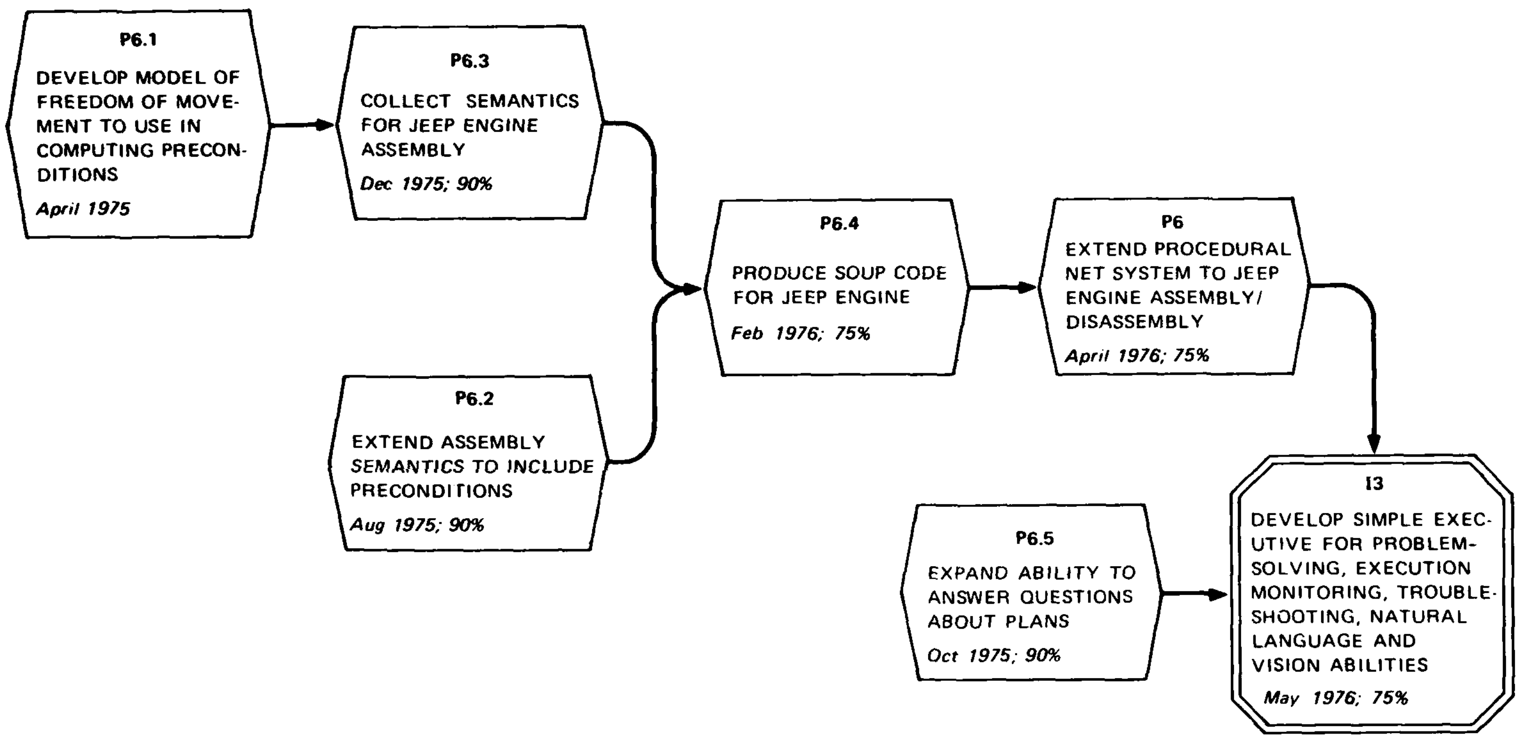


FIGURE 3 ADDITIONAL PROBLEM-SOLVING TASKS

c. Natural Language

The tasks shown in Figure 1 are as follows:

L1 - Collect Protocols of Air Compressor Assembly/Disassembly

To determine the language requirements for the CBC domain, we designed a set of simulation experiments. Dialogs were recorded of human experts aiding apprentices in repair tasks in an environment similar to the one in which the demonstration systems will operate. From these dialogs, we were able to determine the initial language needs of a consultant system and to specify these needs in terms of an initial working grammar, vocabulary, and a set of semantic concepts. A package of concordance programs was built as an aid to enable us to examine word usages in context.

L2 - Design a Semantic Representation System

In this work, now completed, the various representations of semantic information in current use by leading computational linguists were investigated to determine which (if any) was best suited to the needs of the computer-based consultant. Semantic networks were selected because of their flexibility and inherent properties of semantic associativity. However, in their conventional form, networks lacked convenient mechanisms for encoding quantification, distinguishing between alternative "worlds," encoding processes and representing abstractions. To overcome these difficulties, a variation into units called spaces has been developed and is described in (G. G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," SRI Artificial Intelligence Center Tech. Note 105, Menlo Park, California (June 1975).

L3 - Design System for Simple and Ellipsis Resolution (Dialog Context Only)

Examination of the CBC protocols revealed extensive use of anaphoric reference (both pronouns and determined noun phrases) and ellipsis (partial utterances that can be filled in from context). An initial dialog package handles only discourse phenomena that can be resolved by looking at dialog history. References and ellipsis that require a model of the task for resolution are not handled. In addition, because a task model is not yet part of the system, the dialog history is only linearly structured. As a result, only the previous utterance is consulted in working on resolution. The kinds of references that are resolved are limited; for example, only unmodified noun phrases are handled (i.e., "the

bolts" but not "the small bolts". The interface with semantics and the requirements of a more complete dialog system are well specified.

L4 - Build a Semantic Net System Having Multiple Partitioning

Based on the design developed in L2, a set of routines for performing basic network operations will be implemented. These routines will provide facilities for creating and destroying nodes and arcs in a semantic network. Facilities will also be provided for subdividing the net in multiple ways and for assigning arcs and nodes to the spaces composing each division. One division will be used for encoding quantification while another will organize the network into context regions for purposes of anaphora resolution. The spaces of each division will be organized into a hierarchy resembling the hierarchy of QLISP contexts. Retrieval routines will be implemented which (like QLISP) restrict their attention to particular spaces or branches of the space hierarchy.

L5 - Adapt Paxton Parser to Handle Text Input

To simplify the initial development of a natural language input system for the CBC, the problems incurred in working from speech input have been postponed. However, to facilitate the eventual conversion to speech input, we will use the SRI-SDC speech parser and control structure. This speech system, originally implemented on the SDC's IBM 370, has been translated into INTERLISP and is currently being optimized for text input.

The parser has many capabilities that allow it to handle the inherently muddy input of normal speech. These capabilities (e.g., being able to parse small chunks in either top-down or bottom-up mode, working either left-to-right, or right-to-left, or from the middle out) are not required for text processing, but will be needed when the CBC converts to speech input.

L6 - Develop Rules for Translating Parsed Utterances Into Semantic Net Form

While the parser of L5 will provide the mechanism needed to interpret the syntactic characteristics of an input string, it must be supplemented with a system capable of forming semantic interpretations of inputs. The semantic composition system provides this capability by indicating how more complex semantic structures may be built up from simpler structures. Ultimately, the simplest semantic structures are associated with individual words stored in the lexicon maintained by the parser. For each input utterance, the output of the semantic composition

routines will be a network structure encoding the meaning of the input and of each syntactic component used by the discourse routines.

L7 - Design and Implement Preliminary Retrieval System Based on Matching

The routines of L4 provide only rudimentary access into the semantic network. To answer questions, a retrieval system is needed which is capable of manipulating not only the structural form of the net, but also the semantic concepts and interrelations that the net encodes. As a preliminary attempt to create such a retriever, a system will be implemented that embodies a basic knowledge of the set-subset and set-membership relations and the conventions of deep semantic cases. These abilities will allow the retriever to interpret the content of the semantic net at face value (i.e., without being able to perform deductions or use theorems and rules encoded in the network). The retrieval system will accept query structures such as those built by the composition semantic routines of L6 as inputs, and will output pointers to structures in the network data base which are the answers to the input queries.

L9 - Build Preliminary Natural Language Q/A System

Combining the work of L1 through L7, a preliminary natural language question answerer will be built. This system will accept text input queries stated in the "base grammar" and translate them into network structures. In this translation process, the anaphora routines of L3 will find the referents of pronouns and definitely determined noun phrases, but will be limited only to the use of previous input sentences and responses with no appeal to the task environment. While multiple "verb" inputs may be translated into their network representations, the answer retrieval system will be capable of responding only to single "verb" queries (such as "where is the wrench?"), since the retriever will not yet have the ability to combine multiple relational concepts (as in "where is the wrench that I used to tighten the bolts?"). Output from the system will be "YES" or "NO" or a formal statement encoded in network notation. Since jeep protocols (L9 below) will not yet be available, questions will concern tools and simple parts such as nuts and bolts.

L9 - Collect Protocols of Jeep Engine Repair

Work analogous to that in L1 is to be done for repair tasks on the jeep engine. The discussion of new kinds of tasks will entail the expression of new concepts. This

extension will require additions to grammar, vocabulary, and the set of semantic concepts handled by the system. Additions to the discourse capabilities are also anticipated.

L10 - Interfacing State-of-the-Art Speech Understanding With CBC Domain

Up to this point we have postponed considering the problems incurred in working with actual speech input. Now it will be necessary to look carefully at the problems of compatibility between the existing text question answerer and state of the art speech understanding systems. The ARPA 5-year program will be almost complete and this will be a good time for us to look at what is possible with speech input. We will interface the CBC question answerer with the speech system being developed jointly by SRI and SDC and use the resulting system for testing speech input on a limited scale.

L11 - Develop Early English Generation System

To make the output from the question answerer usable by the apprentice, an English noun phrase generator will be constructed which generates descriptions of objects (other than event and relational objects) from their encodings in the semantic network. This preliminary generator will be insensitive to context considerations and will make no use of anaphora.

L12 - Develop Semantic Theory of Processes and Integrate With Task Model

Many of the actions in the CBC domain are actions that cause continuous change of state (e.g., tightening is a continuum that starts at "barely attached" and runs through "fully tightened"). The procedural net provides an internal representation of these kinds of actions useful for planning. A way of representing these actions so that the system can discuss them with the user must be designed. This representation will provide an interface between the natural language input and the task model representation. Basic research on the representation of processes must be done. As a first step in incorporating process semantics in the language system, a system that depends on knowing where the user is in the task structure will be built. This system will guide the user through a procedural net, demanding strict adherence to the prespecified plans. The user will be allowed to determine the depth of detail of a task description, but will not be allowed to take the initiative in choosing the next steps to be performed. At this point, the

ability to model processes will be limited to talking about actions that actually happened or are to happen. The system will not be able to discuss alternative possibilities.

Once this representation is designed, we will be in a position to use task information in the discourse routines--i.e., references and ellipsis that require reasoning about the task for resolution will be handled. In addition we will begin designing a model of the apprentice which takes into consideration various levels of ability and understanding of the CBC tasks.

L13 - Extend Retrieval System

Because of its limited ability to manipulate concepts in the semantic network, the answer retrieval system is expected to be the weakest component of L8. To strengthen the question answerer, the retriever will be extended by adding facilities for the manipulation of quantified statements, deduction rules, and categorical information. These additional capabilities will allow the retriever to compute answers to questions that are not stored explicitly in the network data base.

L14 - Build an Extended Q/A System

With the developments of L9 through L13, the capabilities of the question answerer of L8 will be greatly extended. The new system will accept input text containing both statements and queries concerning the jeep task domain. With the support of other CBC systems and using the ability of L12 to follow procedural nets, the language system will be capable of guiding the user through preplanned task procedures. (But the system must maintain the initiative in order to keep its place in the task structure.) With the procedural net available, the anaphora-resolution routines will be able to resolve references to the task environment that are not resolvable from the discourse history alone. The improved retrieval system will be capable of handling multiple "verb" queries and of deducing (some) answers not explicitly recorded in the data base. Noun phrase answers will be processed by the generator and output in English (either as text or through the speech synthesizer). Commands and instructions bsed on the procedural net will continue to use canned output.

L15 - Add Acoustic Processing Capability

Before conversion to speech input can occur, the hardware and phonetic level software necessary to do acoustic processing will have to be acquired.

L16 - Extend Process Semantics

In order to have a system capable of allowing the user to take the initiative in task performance, the process semantics must be extended to handle user queries about possible results of actions. That is, the ability to answer hypothetical questions about processes must be added. The ability to model various possible states of processes will also be necessary for determining where the user is in the task when he has taken the initiative and is reporting back on some new state.

L17 - Extend Retrieval System to Understand Process Knowledge

The retrieval system of L13 will be competent in the manipulation of static facts, but will have little or no ability to manipulate process knowledge. To respond effectively to user inputs regarding processes, the retrieval system must be extended to include an ability to interpret and manipulate procedural nets. Further, the retriever must have models of action verbs that are more abstract in character than the action specifications of the procedural net. The interaction of abstract event concepts and concrete plans is not yet clear, and the extension of the retrieval system into the area of process knowledge will involve basic research into the nature of process semantics.

L18 - Extend Generator's Grammar

Output from the computer-based consultant will begin to sound "natural" only when a variety of syntactic constructions are available and when human-like use of anaphora and ellipsis is incorporated. Building on the generator of L11, the syntax will be extended to include complete sentences that approach the parser's input capabilities in complexity. This extended syntactic ability will be augmented by a response control module that will regulate the use of anaphora and ellipsis, appealing to both the dialog history and the task environment to condense output. This extended generation ability not only will be useful for output, but, with the eventual conversion to speech, will also be useful in parsing by predicting likely inputs through context controlled generation.

L19 - Design System for Speech Prediction in CBC Domain

One of the requirements of a speech input system is an ability to obtain expectations concerning an incoming utterance from the semantics and the discourse components of the system. Speech input is sufficiently noisy that some ability to predict even categories of what may be said is helpful in limiting the search required for processing inputs. The semantics can make predictions by

knowing what compulsory information is missing from the verb phrases. The discourse routines can supply task context and also detect what missing information cannot be filled in from context. We will incorporate some predictive ability in the semantic and discourse routines for eventual use with speech input, and will test them in the text environment.

L20 - Build Extended Language System

Adding the capabilities of items L16, L17, and L18 to the system of L14 will produce a natural language configuration capable of handling the bulk of the natural language needs of the computer consultant. The system will process all linguistic outputs from the consultant as well as all text inputs. By appealing to the procedural net task model and by processing both inputs and outputs, human-like uses of anaphora and ellipsis will be made within the restrictions imposed by the syntax. The system will, however, not yet process continuous speech, handle syntactically or semantically ill-formed expressions, understand examples and analogies, or parse utterances expressed outside the basic jeep syntax.

L21 - Convert Part of Q/A System to Speech Input

At this point we will have developed a natural language system for text input capable of handling the subset of English occurring within the confines of task-oriented dialogs in the Jeep/workstation domain. In addition we will have tested predictive strategies for guiding a speech parser and will have added acoustic input capabilities. The final conversion to speech input will require developing acoustic and phonetic descriptions of the words in the CBC lexicon, and integrating the acoustic input capabilities with the input-mode independent modules of the natural language system. Successful integration will make stringent demands on prediction capabilities and will require testing, refining, and tuning of the system. We intend to make a start on this task during the CBC project.

We estimate that the total effort needed for the Natural Language Tasks will require a rate of about two persons per year aided by some extra programming help.

d. Troubleshooting

The tasks shown in Figure 1 are as follows:

T1 - Explore Methods for Machine Diagnosis

A review of the literature disclosed several approaches to diagnosis. These included: (a) combinatorial methods that use special test inputs for the diagnosis of combinational logic circuits, (b) information theoretic methods that seek to isolate single faults with a minimum number of measurements, (c) dynamic programming methods that take the cost of tests into account, (d) decision-theoretic methods such as sequential Bayes' procedures that take probabilistic relations between faults and evidence into account, and (e) rule-based methods that allow incremental growth by the transfer of new rules from experts to the system.

The latter approach, exemplified by Shortliffe's MYCIN program, seemed most appropriate as a starting point. MYCIN uses judgmental rules that go from premises about effects to conclusions about possible causes. A variation based on a hypothesize-and-test paradigm that used strong cause-and-effect rules in place of judgmental rules was designed and programmed. It was capable of diagnosing simple electrical circuits, but limited user interaction to answering "yes" or "no" to specific questions.

T2 - Develop a "Simple" Rule-Based Diagnostic System

Diagnosis of larger systems requires both cause-and-effect and judgmental rules. It appears that these styles can be combined nicely, using concepts related to MYCIN and Reddy's HEARSAY II system. The chief problem to be addressed in this task is incorporating volunteered evidence. This capability implies being able to (a) chain forward from evidence to conjecture good hypotheses, (b) score the hypotheses and select a good one for verification, and (c) decide whether to ask for verifying information or to chain backward to find supporting evidence. The version of a rule-based system to be implemented will use simple strategies to solve these problems, such as assigning probabilities to hypotheses and always working on the most likely hypotheses regardless of other considerations.

T3 - Build "Simple" Fault Diagnosis System for Jeep Engine

Using the available working techniques, a rule-based system will be supplied with rules for jeep engine diagnosis. Use will be made of the sensors available from the ATE/ICE system, as well as questions asked

directly of the apprentice. If necessary, the scope of the diagnosis system will be limited to some subset of jeep engine diagnosis, such as diagnosing engines that do not start.

T4 - Develop System to Give Repair Instructions

We will develop a subsystem that accepts the output of the diagnosis subsystem and then gives the apprentice advice about how to repair the fault. Initially, this subsystem will handle only remove and replace types of repairs, but ultimately we hope to be able to extend it to repairs involving adjustments, alignments, etc. The system will make heavy use of the procedural net subsystem for plan generation.

T5 - Extend Fault Diagnosis to Some Other Major Part of Jeep (e.g., Transmission)

A danger involved in any investigation centered on a specific problem is that the resulting techniques may not be extendable. Thus, this task, it is hoped, will both demonstrate the generality of the methods that we will have developed, and serve as an incentive to keep our earlier work from becoming too specialized. This task will also serve as an opportunity to upgrade the simpler system designed for engine diagnosis, and to incorporate other developments in the use of models and general diagnostic knowledge.

T6 - Integrate Troubleshooting Subsystem

The diagnosis and repair parts of troubleshooting will be integrated into a single package that will interface smoothly with the rest of the demonstration system.

Figure 4 shows some important subsidiary troubleshooting tasks. These are described as follows:

T6.1 - Develop Ways to Use a Model to Make Deductions

When a physical system is well understood, an accurate model such as a circuit model may be available. The model can be used to predict the results of tests under various hypotheses, and to draw conclusions from various combinations of evidence. In principle, one might be able to derive rules automatically from the model, i.e., to use the model in a "compiled" rather than an "interpreted" fashion. The intent of this task is to investigate these kinds of questions and to identify effective ways to make use of models.

T6.2 - Explore Ways to Exploit General Diagnostic Knowledge

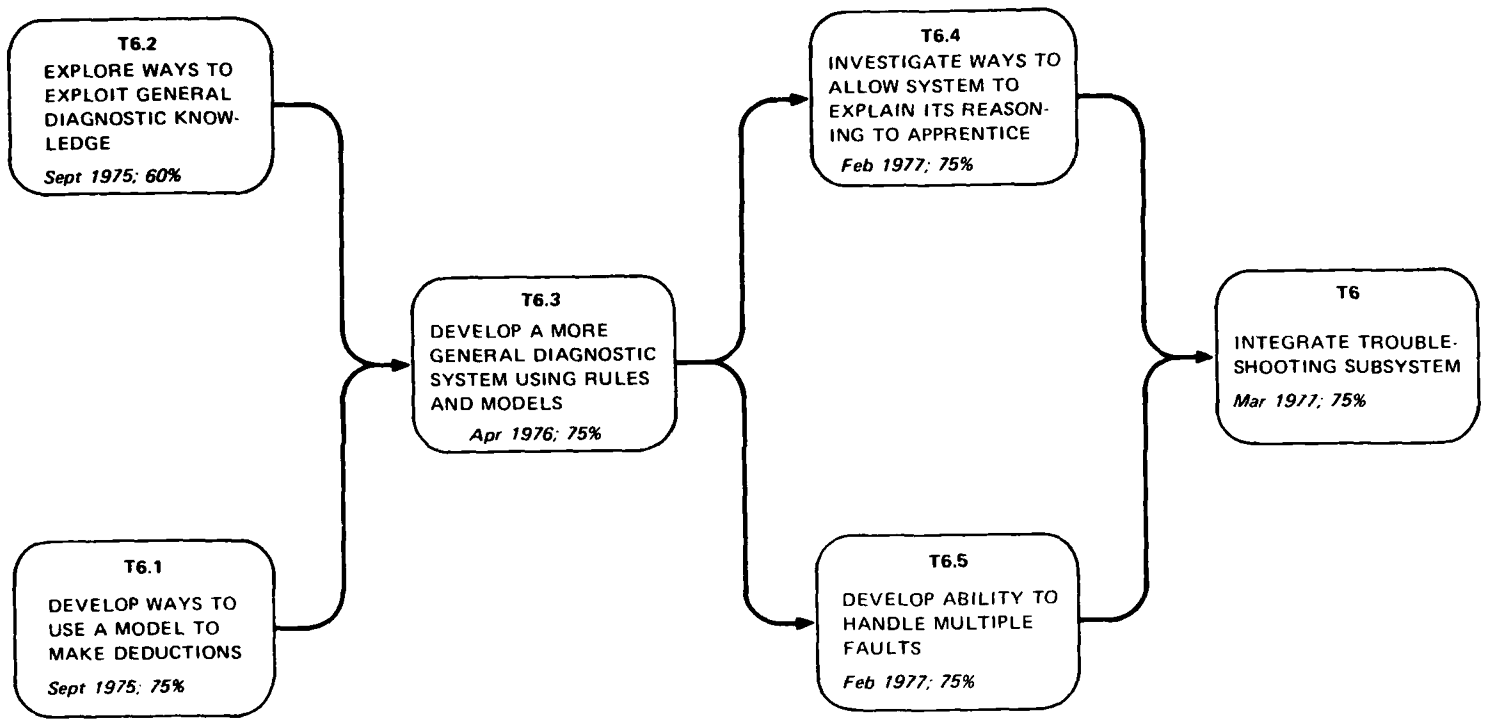


FIGURE 4 ADDITIONAL TROUBLESHOOTING TASKS

Rules that are tied closely to models tend to be specific to the piece of equipment under consideration. There are also more general kinds of rules--ranging from laws of physics to rules of thumb--that are more generally applicable. These rules involve variables, and are often sufficiently vague in their generality that it becomes difficult to know how to apply them effectively in a given situation. Nevertheless, for the purpose of generality, it is necessary to find ways of exploiting general knowledge, which is the goal of this task.

T6.3 - Develop a More General Diagnostic System Using Rules and Models

The separate investigations of combining cause-and-effect and judgmental rules, using models, and exploiting general diagnostic knowledge may not lead to a unified approach to diagnosis. The goal of this task is to integrate the best of these ideas so that various strategies can be fitted together and be experimentally evaluated.

T6.4 - Investigate Ways to Allow System to Explain its Reasoning to Apprentice

One of the major advantages of explicit rule-based systems over implicit systems such as decision trees is that their behavior can be understood in terms of the rules that are being applied. This advantage provides the opportunity of having the system give explanations for its conclusions, thereby increasing the apprentice's confidence in the system. Similar advantages have already been exploited with our procedural nets, and should be a part of the diagnosis system also.

T6.5 - Develop Ability to Handle Multiple Faults

The nature of the diagnosis problem for a system that has been working recently is quite different from that for a system that has undergone major disassembly, modification, and reassembly. In the former case, one can expect only one or at most a few, probably not strongly interacting, problems; and models can be assumed to be largely correct. The strategies change considerably when several faults can be present, or when there are major discrepancies between models and reality. The goal of this task is to investigate these problems and to develop methods that can handle common kinds of multiple faults.

We estimate that the total effort needed by the Troubleshooting Tasks will require a rate of about two persons per year.

e. Hardware Interfacing and Facilities

As shown in Figure 1, the tasks that are pertinent to interfacing and provision of hardware are as described below:

H1 - Complete Interface Between PDP-11/10 and CBC I/O Equipment

The desired state of the computer environment is as shown in Figure 5.

The VOTRAX is a voice-synthesis device which, given a stream of digital data representing a sequence of fundamental speech sounds (phonemes), will reproduce these sounds as intelligible speech. This device is built to be interfaced like a teletype and therefore merely plugs into any teletype port on the PDP-10.

The PDP-10 and PDP-11/10 communicate over the ARPA network via their respective ARPA interfaces. The PDP-11/10 then directly controls and senses the remaining equipment.

The VIP-100 is a commercial device that can be trained to recognize isolated words (or short phrases) from a vocabulary of 64 words (or phrases).

Camera controls (focus, zoom, diaphragm, and color wheel) and the camera pan/tilt head are controlled by digital signals from the computer, and their positions are sensed with a multiplied A/D converter.

Currently, the ARPA interface, A/D converter, rangefinder, and pan/tilt head are all interfaced and operating. Still remaining are the camera controls and the VIP-100. Since the VIP-100 is to be shared between two projects, it will be switchable between the PDP-11/10 and the other project's PDP-11/40.

H2 - Prepare for Changing Task Environment to Jeep

Since it is being proposed to use a jeep as a demonstration vehicle, a great deal of work will have to go into providing a suitable work environment. A basic question to be answered is "How much of a jeep?" Probably for some time to come, a representative subsystem, possibly just parts, will be enough to satisfy the research needs. Possibly something like the engine-transmission-cooling system supplied on a suitable mount would serve.

In any case, a suitable workspace will have to be provided, equipped with appropriate tools, workbenches, hoists, and the like. If it is desired to actually operate the engine, then we will have to be careful in

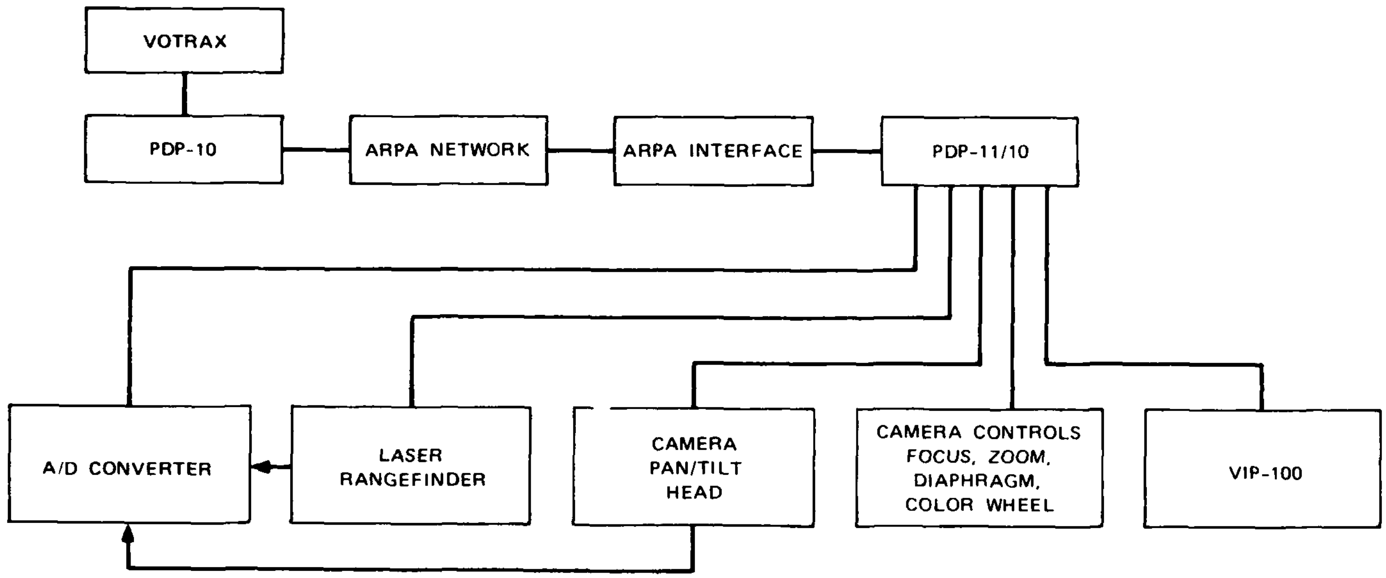


FIGURE 5 CBC PROJECT EQUIPMENT AND COMPUTER CONFIGURATION

the design of the room to provide adequate safety precautions from the viewpoint of possible fire, explosion, exhaust fumes, heat build-up, and other hazards. Noise could also be a problem.

Appropriate repair and maintenance manuals will have to be ordered along with relevant diagrams and parts lists. Any special tools, jigs, or fixtures peculiar to jeep maintenance will also have to be ordered.

H3 - Procure Jeep and ATE/ICE System

Since there may be a fairly long lead time in obtaining a jeep, we should request one as soon as we can. We can probably get both the jeep and the ATE/ICE system from a relevant Army agency, or possibly it could be more advantageous to buy them on a GSA contract to take advantage of the government's quantity prices, and at the same time, eliminate some of the delays of processing through too many government agencies. In any case, we will be coordinating all of this with ARPA.

H4 - Interface ATE/ICE Transducer System with our Diagnostic System

As soon as we receive the appropriate documentation on the ATE/ICE system, we can begin designing and building the required interface with our PDP-11/10. We will also have to decide what makes the best sense in the physical placement of the various components. Should we leave the PDP-11 in the computer room where it now is? Should it be moved to the vicinity of the jeep room? Would it be better to separate the transducers and the computer of the ATE/ICE system so as to place that computer in our computer room?

We estimate that the total effort required for the hardware tasks described above will require a rate of about one enginer per year plus about 1/4 to 1/2 technician per year.

f. Vision

The tasks shown in Figure 1 are as follows:

V2 - Locate Air Compressor and Major Parts by Laser

A simple set of procedures has been implemented for locating known, isolated objects, using range data and structural models.

V3 - Complete Simple Fixed-Camera Model

The camera will be modelled and calibrated so that we can set parameters correctly to take a picture and precisely determine the geometrical relation of scene to image.

V4 - Complete TV Instrumentation

An essential step is to ensure that the TV system is as good as we can make it. We need to readjust the camera, set up additional lighting, and especially, modify the (10 year old) digitizer. We are in great need of higher dynamic range, lower noise, logarithmic encoding, and more gray levels.

V5 - Measure and Calibrate Camera

Measurements will be made of the characteristics of the TV system so that we know the sensor-dependent limitations of the image (e.g., vignetting by the zoom lens).

V6 - Digitize Photographs of Jeep Parts

In parallel, and so that our research shall be as unimpeded as possible by the limitations of the TV equipment, we shall be taking photographs of various objects for digitization on a high quality scanner.

V7 - Study Visual Properties of Jeep Domain

The visual appearance of surfaces of various shapes and reflective natures will be studied so that we can form models of illumination and reflection appropriate to the domain.

V8 - Assess Segmentation Techniques

The prevailing techniques for low-level scene segmentation will be assessed to determine a basis for our system. The machinery domain clearly requires line-finding and region-finding in some combination. We will have to synthesize a new technique from what is currently known and from the results of our domain studies.

V9 - Correlate Model and TV Picture

Some special-purpose procedures will be written for refining the position estimates obtained from range data by correlating TV pictures with appearance predicted from structural models.

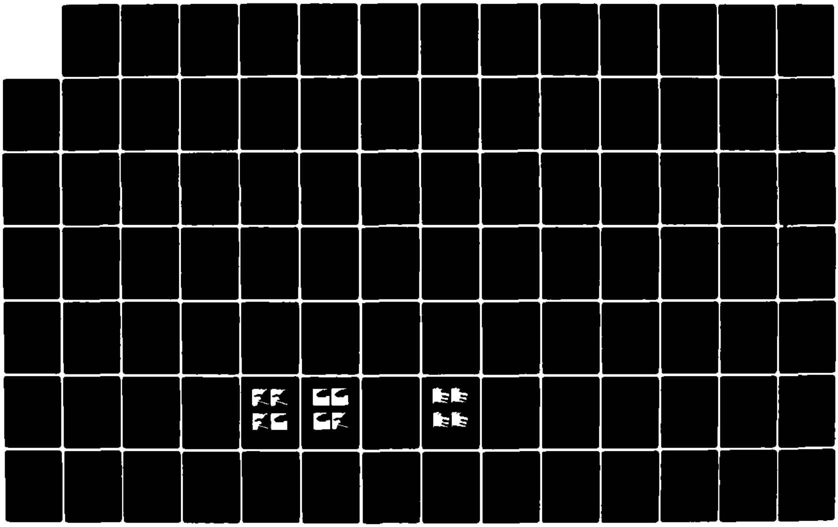
V10 - Develop a Simple Picture Segmentation System

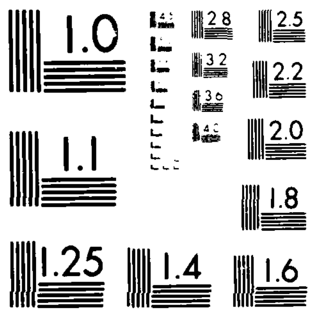
We shall design and implement a low-level segmentation process that can combine information about edges, shading, and lighting and which is capable of being guided by higher level information about structures.

V11 - Develop Interactive System for Describing Equipment in Geometric Terms

AD-A122 473 ARTIFICIAL INTELLIGENCE -- RESEARCH AND APPLICATIONS 2/3
(U) STANFORD RESEARCH INST MENLO PARK CALIF B RAPHAEL
JUN 76 DAHC04-75-C-0005

UNCLASSIFIED F/G 6/4 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Describing new equipment in geometrical terms is currently a tedious and time-consuming task. To enable us to easily add new components and subsystems to our models of the jeep, we will develop interactive methods for building up models and displaying them as they are constructed to verify that they are correct.

V12 - Locate Parts by Laser, Guided by by Models

The special-purpose hand-coded procedures will be generalized so that the system can decide for itself the necessary strategy for locating a known isolated object from range data, given the structural model.

V13 - Locate Parts by Laser and Verify by TV

Location of parts from range data and verification of predicted appearance on TV will be combined into a single system.

V14 - Recognize Parts by Laser and Verify by TV

Recognition of parts from a small set, using range data and TV verification, will be accomplished.

V15 - Extend and Refine System for Recognition

The laser and TV systems will be unified into one coherent system that uses range and TV data to recognize and locate parts in complex scenes. Possibly some ability to detect incorrect assemblies may be incorporated.

V16 - Extend and Refine System for Recognition

A general system will be put together that uses TV and laser data in scene analysis. It will be able to identify and locate objects, and check for correct assembly.

We estimate that the total effort needed for the Vision Tasks during the period of the current proposal (October 1975-January 1977) will require a rate of about one person per year.

4. Comments on the Importance of CBC Systems for Maintenance

This section presents the case for the need for computer-based expert systems, particularly in the area of maintenance.

a. The Need for Automated Expertise

First, we observe that there is currently a critical shortage of human expertise in several key military endeavors. Furthermore, it is likely that this shortage will become more critical in the future. Factors contributing to the shortage are:

(1) The All Volunteer Military Force. With the end of the draft, it has been difficult for the services to obtain personnel possessing the needed technical skills or even to obtain personnel who can be trained to the needed levels. Too many of today's volunteers have not continued their regular schooling past the ninth grade.

(2) Rapid Turn-Over of Personnel. In order to attract volunteers, enlistment periods tend to be short. Also, a large number of personnel who do become skilled change over to civilian jobs rather than re-enlist. Frequently, for example, personnel either leave or report to a ship in mid-cruise.

(3) Growing Complexity of Equipment. The need for expertise is growing because the services are constantly using more complex equipment. The skill levels needed to operate, modify, and maintain this equipment are increasing, and yet the proportion of personnel who have been or can be trained to these higher skill levels is decreasing.

(4) The Nature of Certain Missions. The space-age defense establishment operates a large and growing number of complex sites which, by their nature, can be manned by only a small number of personnel. Examples are ballistic missile warning stations, nuclear submarines, and--possibly in the future--space stations. In these sites there is a particularly acute imbalance between the need for expertise and the supply of personnel who can provide it. Furthermore, in sites like nuclear submarines, the expertise must be on-board because of the obvious restrictions on communications.

(5) Psychology Factors. Certain frailties such as fatigue and boredom seriously impair the effectiveness of even highly trained personnel on many jobs. Duty periods must therefore be reduced to obtain acceptable performance levels, and this reduction creates a need for additional personnel.

Several measures are suggested to alleviate the expertise shortage. The services could attempt to provide more human experts through expanded recruiting and additional training. We suspect that these routes are nearing saturation. Ultimately the existing experts must be made more effective by multiplying their availability through various job performance aids.

Conventional aids include manuals, video and film clips, sound recordings, and the like. Such aids can help alleviate the problem, but have several disadvantages that limit their utility. The major disadvantages are:

(1) They are static. A manual, once written, stays the same until it is rewritten or supplemented. Keeping manuals up to date is expensive, and in fact, is never achieved in a wholly satisfying manner.

(2) Most manuals are written to satisfy the lowest common denominator of skills--that is, they are written for the relatively unskilled. Therefore, they are usually bulky and unwieldy. Relatively more skilled users find the manuals tedious and do not use them. If manuals were written for higher skill levels, then less skilled personnel would not be able to understand them. In short, the same manual cannot be matched to all users.

(3) They are not interactive. A manual cannot engage in a dialog with the user. It cannot suggest specific advice matched to the user's immediate problem. It cannot ask the user questions for more information about the problem, or even answer a user's question directly.

(4) Many volunteer servicemen do not have sufficient reading skills to make effective of use of written manuals.

It is our conclusion that the need for expertise cannot be met satisfactorily by any combination of increased education or conventional job performance aids. The need can only be met by producing automated experts--computer systems that can engage in dialogs with their human users to help them perform increasingly complex tasks.

Besides overcoming the disadvantages of manuals and recordings listed above, a computer-based expert would have the following important additional advantages:

(1) Reproducibility. Computer programs can be reproduced as often as needed; thus, the supply of expertise can be increased to meet demand.

(2) Updating. Modifications to the expertise in a computer system can be made by changes and additions to the program.

(3) Communication. Expert systems can be quickly transmitted to wherever they are needed via computer network systems. In particular, updated systems can be made immediately available.

(4) Record keeping. Automated expert systems can be integrated with computer record systems to update maintenance files on specific military items, types of malfunctions, and the like. Present record gathering procedures do not function as well as is desired; the introduction of computer-based experts at the sites where data are gathered would thus have the important side effect of improving data gathering.

(5) Computer assisted instruction. Automated expert systems can be used in an instructional mode to help train technicians.

b. Maintenance of Military Vehicles as a Task Area

Notwithstanding our goal of developing fundamental and widely useful technology, we must select some particular task area in which to pursue the research. We want a task area that is important in its own right as well as one that serves as a typical representative of a wide variety of applications.

We think that maintenance of military equipment is such an area. There can be no doubt about the importance of maintenance. Furthermore, there is a growing awareness among the military, Congress, and the general public about the size and cost of the maintenance problem.

Estimates vary about the amount of money and effort devoted to military maintenance. A reasonable number is probably about $25 billion per year. A 1968 study concluded that it would be possible to save $33 million per year in maintenance costs of Army combat and transport vehicles alone by automating certain maintenance procedures at the organization level. * Sizeable percentages of the enlisted personnel in each of the services are engaged in maintenance activities.

Even with all of these resources devoted to maintenance, it is still not performed well enough. Incorrect diagnoses and repair procedures cause additional expense for rework and can interfere with

----

* R. J. Brachman, "Economics of Introducing Automatic Diagnostic Equipment to Organizational Maintenance," Memorandum Report M68-33-2 (Revised 15 April 1969), U.S. Army, Frankford Arsenal, 15 August 1968.

the performance of missions and endanger lives. Commanders must take into account the fact that a certain percentage of equipment will be "down." Thus, to meet operational requirements, the total capital investment in equipment is larger than would be necessary if equipment malfunctions were diagnosed quickly and accurately and repaired promptly.

Obviously, these problems can be partially solved by using equipment of increased reliability, and undoubtedly more progress can be expected in reliability engineering. But we will always be faced with the necessity for repair, and additional expertise in this area can lead to substantial savings and increased effectiveness.

We mentioned earlier the fact that computer based expert systems offer the additional advantage of automated record keeping. This is particularly true in the maintenance task area where it can lead to much more reliable information about materiel readiness. For example, whenever a vehicle is submitted for repair, the automated maintenance system will also remove it from the list of operational vehicles until it is redeployed.

C. Prognostics for Vehicle Maintenance

by Steven H. Johnson

1. Background

Interest has recently been shown in the concept of predicting failures in vehicles and vehicle subsystems as a means of maintaining materiel readiness. This concept holds the promise of cost effectiveness and increasing vehicle reliability during missions, since major breakdowns would, it is hoped, be identified before they occur. Related to the concept of prognostics is that of incipient failure identification. This latter idea deals with detecting minor malfunctions and calling attention to them before they lead to catastrophic failures. This is, therefore, similar to prognostics.

In the field of artificial intelligence, there exists the necessary technology for making predictions. It seems appropriate at this time, then, to link prediction capability to the significant problem of military vehicle maintenance. Further, it is apparent that the technology necessary to accomplish vehicle prognosis for certain types of failures can be made available in the near future. For these reasons, we are proposing to devote part of the project effort to studying and developing vehicle prognostic systems. This topic also appears to be a natural complement to the vehicle-oriented Computer Based Consultant.

This subsection discusses the goals and rationale behind vehicle prognostics, mentioning the current status of prognostic and diagnostic systems, and outlining a work plan for this part of the project.

2. Goals

The plan for our prognostics effort, discussed in some detail below, will be aimed at accomplishing the three major goals listed here:

We will assess previous work in this field with a view toward identifying the most important areas of prognostics and preventive maintenance in which AI techniques could make a significant contribution. The assessment will be made in terms of frequency and importance of failures and the real cost thereof. This assessment will include a present-day evaluation of the costs and benefits to be realized by prognostic systems.

This plan is expected to focus on one or more existing vehicle problems that are both amenable to failure prediction and critical to materiel readiness. This plan would be implemented on a small scale to demonstrate feasibility. We will concentrate on problem areas identified in Part B-1, above.

We recognize several areas crucial to implementation of prognostic systems. We intend to address those critical topics and produce practical recommendations.

3. Rationale

There are a number of reasons why the subject of failure prediction for vehicles should be pursued, including such topics as rapid turnover of personnel, growing complexity of equipment, and the importance of specific missions. At this point, it seems worthwhile to briefly add to the rationale behind prognostic systems.

The concepts of problem identification before catastrophic failures occur and maintenance of vehicle readiness imply benefits to the material users. First, if problems can indeed be predicted, it is likely that the cost of replaced parts or assemblies will be far less than that of failure repair. At the same time, the labor hours required and necessary mechanics' skill should be reduced. Second, enhancing vehicle readiness suggests less vehicle down-time. To keep a given number of vehicles in service, then, requires a smaller total number of vehicles. Third, less tangible, but no less important, benefits in the form of user confidence in the vehicle and mission security can result.

No cost projections or analysis of cost effectivess of prognostics have been made. Data are available, however, regarding the present yearly costs of maintenance of military vehicles. The magnitude of maintenance expense suggests strongly that prognostics be considered as an avenue for reducing maintenance costs. In Brachman*, examples are given comparing costs of individual bearings to the cost of the assembly that would have to be replaced if the bearing were to fail. The difference between these parts costs was one to two orders of magnitude. That example was used to show a dollar benefit of incipient failure detection; the same argument can be used regarding failure prediction. The labor required to replace a defective minor component may not be an *order of magnitude* less than that required to rebuild or replace an entire assembly, but tangible dollar benefits are also expected here.

Computerizing prognostics also provides a means for maintaining complete records on individual vehicles, including a running record of important functions such as bearing noise and ignition

performance. This running record, which is similar to periodic sampling of EKG and blood pressure data in prognostic medicine, will allow a statistical history of signals correlated to failures to be compiled. The data could also be related to date and place of model manufacture or specific field use of a vehicle group.

A final reason for pursuing prognostics at this time is that most of the necessary technology to implement such systems is now available.. Early systems, dealing with perhaps the few most crucial historical problems, could be implemented and useable in a reasonable time. Even an early system would use artificial intelligence techniques and could be designed to be an adaptive, evolving program.

4. Status

A great deal of work has been done in recent years regarding vehicle diagnostic systems. A variety of projects in this field have led to the development of on-board sensors and displays; on-board logic; programmable electronics that conduct tests automatically; electronic components for use on board, in the field, or in a depot; and analytical diagnostic software techniques. A considerable amount of literature has been generated in the process of developing this prognostic systems and incipient failure detection.

It is apparent that one of the primary differences between diagnostic and prognostic systems is that a prognostic system must establish and use a valid data base. We recognize that certain diagnostic techniques now incorporate some historical data; in a prognostic system, use of such data is imperative. With the exception of this data base usage, many of the diagnostic techniques are directly applicable. These would include such items as use of similar sensors in both systems, similarity between electronic control and logic hardware, and similar back-up equipment at a depot level. Thus, we expect prognostic systems to get a strong boost from existing diagnostic techniques.

Sensors available for vehicle monitoring can be broken into classes such as on-board sensors, off-board sensors requiring attachment to the vehicle, and off-board remote sensors. These sensors are used to monitor pressures, temperatures, liquid levels, acoustic energy, mechanical vibrations, electromagnetic signals, and the like. Sensors can also be divided into primary and secondary classes. Primary sensors measure a primary quantity, such as oil pressure, that is directly related to such components as the oil pump and main bearings. Secondary sensors are those such as exhaust gas analyzers, which infer from the exhaust gas analysis the operating performance of the carburetor and ignition systems. We suspect that much of the available sensor design and development is immediately applicable to prognosis.

Beyond sensors, present-day technology has additional offerings for prognostic systems. Microprocessors have been developed to the point where they can now be considered for on-board vehicle use. Large scale integration and microcircuit technology will permit packaging of prognostic equipment in a size suitable for on-board use. Statistical techniques, used in conjunction with AI predictive methods, can lead to failure predictions with known probabilities. Computer techniques are now available that can digest input information to continuously expand and update the program's predictive capabilities. All of these factors seem to imply that early prognostic systems could be implemented within a reasonably short time frame.

Engine prognostic systems are now in use by at least five major commercial airlines. Two of the airlines record engine instrument readings manually once per flight. The other airlines record large quantities of parametric data on tape during each flight. In this case, the data are recorded automatically or whenever the flight engineer notes unusual instrument readings. In both systems, the engine data are sent to a center for computer analysis. Trends in data for specific engines are analyzed and reports are produced indicating engine condition and sources of possible trouble. All of the airlines using this form of prognosis feel that the systems are cost effective, allow

them to schedule repairs efficiently, and minimize costly failures. An additional feature to be implemented soon is the in-flight comparison of critical parameters to prescribed limits; if values are beyond allowed limits, data will be permanently recorded automatically.

5. Plan

We have developed the following tentative plan for work on prognostics. Deletions, additions, and amendments to this plan are possible. The reader will note that the first section of this plan is preliminary and deals with the field of vehicle prognostics in general terms. The focus of this section is to arrive at a more detailed program format.

a. Establish a Foundation for Prognostic Systems

A wealth of recent literature exists regarding vehicle diagnostic systems, automatic test equipment, on-and off-board sensors, and related topics. This includes information regarding the status of failure prediction in the commercial aviation industry. We plan to review the available literature to define more closely what is currently available and proven in terms of diagnostic equipment, methods, and sensors.

As mentioned earlier, prognosis depends upon the establishment and use of a valid data base. A data base might include data on large populations of specific types of vehicles and information from isolated vehicles. We hope to determine what is currently available in terms of either type of data base.

In the process of reviewing pertinent literature, looking at data base requirements, and developing a more detailed plan (described next), we intend to make an objective appraisal of the advantage of prognostic systems. We plan to study these closely, so that an accurate prediction of costs and benefits can be made. Past reservations about prognostics may be disspelled in light of technology

now available. It is also conceivable that currently unrecognized problems will make prognostic systems impractical.

The final element of work is the definition of a short-term implementable prognostics program. In as much detail as possible, we intend to define a small scale prognostics program that will exhibit practicality and cost benefits within a short time framework. It is hoped that such a plan would be demonstratable within 12 tc 18 months after contract acceptance. The short-term program will probably deal with a few specific problems on a given vehicle type, and will include suggestions for establishing and updating the data base.

b. Elements of Predictive Systems

Listed below are six elements of prognostic systems. We intend to deal with each of these elements to arrive at specific recommendations on how they should be incorporated. We hope to keep as much of the system as possible on board the vehicle itself, so that the dominant line of communication is between the system and the vehicle operator.

Memory

On-board memory options include tape cassettes, replaceable chips, and field programmable chips. The memory can serve several functions. First, it can be used to store data base information and sensor measurement limits, so that abnormal conditions can be detected. Second, it can periodically record vehicle operating data; such records can be used for evaluating trends in parameters or for later processing by a centralized computer. By replacing or reprogramming either a chip or a cassette, it will be possible to upgrade the system data base as information is compiled.

Controller

An on-board controller will work with the vehicle sensors and data base to take measurements, compare them to limits, evaluate

trends, record pertinent information, and alert the driver to impending failures. It is just now becoming practical to consider ruggedized microprocessors for use as the system controller. Since microprocessors execute programs for incorporated memory chips, it will be possible to easily update the prognostic execution programs, just as it will be to update the data memory elements.

For the purpose of research demonstrations, a minicomputer would probably be used. The capability now exists of condensing many minicomputer operations into microprocessors. Minicomputers are preferable for development work since their programs are easier to modify and debug, especially in evolutionary work.

Sensors

The classes of sensors now available were mentioned above. We plan to review and use as many existing types of sensors as possible, keeping in mind the military requirements for low cost, reliability, ruggedness, and other characteristics. At the same time, we will be alert to the needs for new types of sensors. For example, no sensor is currently available to measure fan belt condition. If fan belt failure proves to be a critical problem, we would spend some time on this topic. (Alternatively, the on-board system could be sensitive to an abrupt water temperature rise.) We will put emphasis on sensors that can be built into the vehicle and continuously connected to the on-board controller.

Display

Display formats will be conceived to interact efficiently with the vehicle driver. Since it is likely that several parts of the vehicle will be monitored by sensors, it is probably impractical to display all values simultaneously. A simple digital display that the operator can switch to the various sensors might be preferred. An alphanumeric display with, say, 20 characters, could display instructions or warning messages. Controller logic could override the

operator-chosen display to show warning or emergency conditions. In extreme circumstances, this visual warning might be coupled with audible alarms or messages. It would also be possible to store messages during vehicle operation and display them to the operator on shutdown, so that he would know what action was necessary before next use. Our general feeling is that any prognostic display must be rugged, reliable, easy to read, and clearly understandable.

Data Base

Development of a valid data base is crucial to the operation of any prognostic system, and is anticipated to be one of the major obstacles to the success of prognostics. After review of currently available data that could be used in the data base, we will develop suggestions for either obtaining an independent data base or upgrading existing information. We plan to consider data for classes of vehicles, and methods of maintaining records on individual vehicles.

Data base information will be contained in the prognostic memory, so that measured values and trends can be compared to prior circumstances. At the same time, the memory can be recording information from its parent vehicle. Thus by periodically replacing the memory element with a new one, fresh information related to vehicle operation can be obtained. This method can be used as a data base bootstrap approach, so that the data base is continuously upgraded based on information taken by the system itself. Depending upon the number of sensors involved and the size of the memory element, it may be possible to exchange memory elements as seldom as once or twice per year.

Central Computer Facility

It is unlikely that on-board prognostic systems could function without intermittent communication with and support of a central computer facility. This support would incorporate the powerful techniques of artificial intelligence. For example, data from a vehicle population can undergo various types of statistical treatment to relate

sensor readings to later equipment failures. It is possible that only combinations of sensor readings will successfully indicate forthcoming problems; and therefore, combinations of readings should be correlated. As a data base is established, probabilities of impending problems can be calculated, thereby indicating the chances that a mission might have to be aborted. All of this information could be processed at a central facility, and condensed for use within the vehicle system, so that on-board predictive capacity is continuously upgraded. We should also not overlook the possibilities of direct contact with a central computer using packet switching radio technology.

The tie between a central facility and an individual vehicle does not necessarily have to be frequent. The main criterion appears to be a continuous flow of information from large vehicle populations to the central facility, so that the data handling and statistical processing just described can go on.

c. Vehicle Problem Identification

In the process of literature review and study of available data, we expect to identify the most critical problems related to materiel readiness. We hope to identify the specific failures that cause the most significant problems, especially with respect to mission success and operator confidence that his vehicle will function properly. We also hope to identify the most common types of failures related to vehicle systems. Obviously, the most significant problems will not always be the most common ones. For example, a bearing failure, although rare, can immobilize the whole vehicle, while wornout windshield wipers are more common but usually just a nuisance. The main emphasis here will be to identify problems that are both significant and common.

d. Predictive Opportunities

We will attempt to identify the specific types of failures that can be predicted. For example, acoustic emissions from bearings are sometimes a predecessor of impending problems. Statistical handling of vehicle population data might show a high incidence of specific component failure when a vehicle has been driven a certain number of hours or miles. This information could be used to warn the driver of problems likely to occur, independent of sensory information. Trends in measured parameters might also precede trouble.

We will compare the list of predictive opportunities with the lists of significant problems and common failures. These three charts will together identify the predictive opportunities that both should be and can be approached. In cases where significant common problems are found but no predictive methods are available, we will try to conceive new prognostic methods. The emphasis of this part of the work will be on pinpointing the areas where prognostics can be beneficially implemented within a short time period.

e. Experimental Work

Experiments will be conducted during the project as suggested by the work and when appropriate. It is not anticipated that any sizeable vehicle population will be instrumented under the proposed effort. However, limited experiments, perhaps using the CBC Project jeep, may be carried out. Further, experiments can be performed on components of prognostic systems. These might include reliability and durability tests of sensors, statistical manipulation of data bases, and mock-ups of vehicle displays. We expect limited experimental work to precede and influence implementation of an early prognostic demonstration system.

f. Long-Term Possibilities

The proposed work on prognostics will emphasize short-term results. At the same time, we will be alert to the possibilities of longer term prognostic methods. These will probably involve more exotic sensors whose data require later interpretation. Recent discussions within SRI have identified topics such as the following, which will be briefly considered in relation to vehicle prognostics.

Thermal Images

Cameras are available which take infrared photographs; gray scale graduations on these images can be interpreted as specific temperatures, with resolution exceeding 1 degree F. Artificial intelligence techniques can be used to scan such images to identify abnormal conditions and localized problems. As an example, an infrared photograph of a radiator could show flow restrictions as local hot spots which could eventually lead to cooling system failure. Comparing successive images to identify trends could also be useful.

Acoustic Interpretation

Multiple microphones could be used to effectively focus their listening ability on a specific component. By comparing acoustic "signatures" from the component at intervals, trends in acoustic images could be identified. Comparing the signatures or trends with data base information could lead to failure predictions. Although recording noise signals is easy, later analysis would require AI techniques.

On-Board Secondary Sensors

A number of secondary sensors have recently been identified for diagnostic use. Interpretation of their data is not yet entirely clear, even in diagnostic systems. Further work will be required to relate secondary sensor output to prognostics. At the same time, secondary sensors will have to be developed to the point of usefulness under on-board operating conditions.

Long-term possibilities such as these will be considered under the project since they will likely be part of a spectrum of near-to-long-term elements of materiel readiness systems. It is also expected that additional long-term possibilities will be identified.

IV DECISION AIDS FOR COMMAND AND CONTROL

by

Earl Sacerdoti, Gary Hendrix, Daniel Sagalowicz, and Jonathan Slocum

A. Introduction

During the past six months we have begun a major new effort in support of the ARPA/IPTO "Command-Control Architectural Testbed" program. The following activities have been completed during the initial six months of this effort:

* We have investigated the data management systems available on the ARPAnet, and have determined that the Datacomputer software, developed by the Computer Corporation of America (CCA), is the current best choice to support the data management needs of the testbed.
* We have examined the Datacomputer's capabilities in detail, and have developed specific recommendations for extensions that are necessary for its use in the testbed environment.
* We have created, in cooperation with the Navy Electronics Laboratory Center (NELC), an unclassified file structure for a set of "status of forces" files that will be extended to provide the initial data base in the testbed environment.
* We have populated this structure with data about 214 ships, and have made the resulting files available on CCA's Datacomputer for use over the ARPAnet by interested researchers.
* We have developed specifications for a preliminary File Access Management system that will provide a robust file access capability in the distributed data environment of the testbed.
* We have implemented an initial version of the File Access Manager based on a robust and efficient interface between INTERLISP and RDC, the TENEX subsystem that provides interactive access to the Datacomputer.
* We have adapted the SRI Speech Understanding System to process textual input. In the longer term, this will serve as a front end to a query answering system about the testbed's distributed data base.

* We have applied a task-specific language interface package, called LIFER, to provide an initial front end to our developing query answering system.
* We have assembled an initial demonstration system that processes English-language queries about a data base of ship characteristics that is stored on CCA's Datacomputer. The system performs comfortably in real time.

Each of these accomplishments will be discussed in more detail below.

B. Selection of a Data Base Management System for the Testbed

When we first started the process of selecting an adequate Database Management System (DBMS) for the testbed, we expected to evaluate the possible candidates against many different criteria having to do with data base size and mode of usage. However, one very important criterion is satisfied by just one system. In the testbed, it is essential that data be accessible interactively over the ARPAnet. Currently, only the Datacomputer, developed by the Computer Corporation of America, meets this criterion. Various other DBMSs will someday be on the ARPAnet. However, it does not appear that interactive ARPAnet access to any of them will be achieved before 1977.

Two systems have so many attractive features that they are worth mentioning even though they are not appropriate at this time for use in the testbed. The first of these is DBMS-10 [1] a CODASYL-type DBMS [2] for the PDP-10 under the TOPS-10 operating system. This system was developed by the Digital Equipment Corporation. We understand that a version of DBMS-10 will be available to run under the TOPS-20 operating system that will be used on the DEC 2040 computers in the testbed. The second noteworthy system is INGRES [3] which is being developed at the University of California at Berkeley. This is a relational DBMS for the PDP-11, and runs under the UNIX operating system. Since a PDP-11 with UNIX will be part of the testbed configuration, INGRES might be a good second DBMS to introduce into the testbed. Interfaces to the ARPAnet for UNIX have been developed by Rand and the University of Illinois. We

have not determined how much extra effort would be needed to build an interactive INGRES-ARPAnet interface.

Although the Datacomputer is clearly the DBMS to use, we nevertheless feel that it suffers some major flaws. Some of these are just errors in or limitations of the current implementation; others are features that are totally missing from the functional specifications of the Datalanguage (the Datacomputer query language). In our experience with the Datacomputer, we feel that we lost about three man-months of effort before we stopped discovering new "limitations" to be circumvented. In the following paragraphs, we attempt to give a catalog of those limitations that we consider the most annoying, or the most serious for the type of usage we envision in the testbed.

* The error messages provided by the Datacomputer are not detailed enough to help the user debug the data used to create or update the database. A typical example occurred often when we attempted to read back a file that had just been created: we would often get the message: "crufty character," without any further information about where it occurred, and what the "crufty character" was. Even if a file is only a few pages long, finding a crufty character in it may take a full day of work.

* Not all variable/fixed length conversions are handled correctly. It is not clear whether this can be considered as an error in the present implementation, or as a more serious fundamental mistake. However, we know that the correction of this error is not trivial. We discovered this error while trying to avoid the "crufty character" problem mentioned above. To avoid it, we decided to create all the files at SRI in a fixed length format, since this would ease the problem of looking at them in printed form, and also would help locate the famous "crufty characters." However, after one week of effort, it appeared that it was impossible to create a variable length file on the Datacomputer by sending fixed length files over the ARPANET. Consequently, we decided to create the entire database in a fixed length format.

* The Datalanguage does not support the notion of priority: there is no way to indicate that a given series of commands should be treated with some level of urgency. This is a severe limitation if the Datacomputer is to be used in an operational environment.

* The level of integrity checking is very limited. In an operational environment, many typographical mistakes are likely to occur when the data base is updated. One way to limit the occurrence of such errors is to introduce some integrity checking: for example, whenever a date field is being written, the DBMS would check whether the year, month, day, hour and minute values are reasonable. A value of 13 for a MONTH field, for example, should be rejected. Many DBMS's have facilities to do these types of elementary checking. The Datacomputer does not.

* The separation between the physical description of the files and their logical usage (the traditional distinction between physical and logical schemata) is not as complete as it might be. In particular, if one wants to read or update a file easily, then the field names which appear in the physical description of the file must be used. In an operational environment, many different programs will access the files. Requiring all the programs to use the same field names is a severe restriction. Many DBMS's do not have this limitation. We suggest that the Datalanguage be extended to accept the bindings between port fields and file fields to be done by physical position as well as by name, as the user's option.

* The current user manual does not specify what happens in case of simultaneous accesses by different users to the same file. We understand that a major development effort is now underway in this area. An early release of a description of the functional capabilities to be achieved would be desirable.

The Datacomputer was designed for managing very large files whose size is on the order of billions of bits. Thus it is not especially well-suited for rapid, frequent, shared accessing and updating of medium-sized files of on the order of millions bits. We think it is unlikely that command and control data bases that are queried interactively will exceed this size in the near future. Therefore, the interface between user programs and the database should probably be performed by a front-end processor, using files that are local to the processor. The only use of the Datacomputer would be to archive or dearchive the files to be used locally. However, in the Testbed there will be no local DBMS facility to access these files other than the Datacomputer itself. This will probably not be a problem until the number of files and users grows significantly, and the computational demands on the Testbed system become heavy.

C. An Unclassified Replica of a Command and Control Database

We have created, in cooperation with the Navy Electronics Laboratory Center (NELC), an unclassified file structure for a set of "status of forces" files that will be extended to provide the initial, classified data base for the Testbed.

The data fields were selected by NELC, based on several databases in current use by the Navy. Classified fields have been suppressed. The data base has been populated with data generated by NELC that was obtained from unclassified sources or generated to fit a fictitious but plausible scenario.

The database is composed of six files, which are described in general terms below. The precise description of both the field formats and their meaning is given in Appendix III, which is in fact a DataLanguage description of the files. Appendix IV gives the field values for all the records in the data base.

1. Dynamic Ship Characteristics

The first file is SHIPFILE. It includes information which is specific to each particular ship. Some of this data is static identity information such as the ship name, hull number, international radio call sign. Most of the information contained in this file is dynamic in nature. Such is the case of the state of readiness, and the reasons of non-readiness for U.S. naval ships; the cargo type and quantity, the number of passengers of merchant ships. The most current value of each of these elements is stored in this file. The interpretation of the values of some fields is dependent on other information in the file. This is the case of the ship position: if the ship is in a convoy, its position is found in the corresponding convoy record, and not in the ship record itself. If it is a US naval ship, and its position is not in the ship record, then its position is the same as the ship carrying the organization indicated by OPCON. Its position may thus be found in the corresponding ship record.

In the physical data base, we could have repeated this location for each ship; however, this would have created a great deal of redundancy. It is now common to consider that a database should avoid redundancy as much as possible, and therefore, we decided to keep the position fields empty in such cases as explained above.

All the fields in this file have a fixed length: this was done because it is much easier to both create and modify the records when they are in a fixed format: since this file is essentially dynamic, we thought that the ease of updating was a much more important factor than the size of the file.

2. Track History

The second file is the track history file, called TRACKFILE. In this file, each record represents either a reported past position, or an estimated future position of a ship or a convoy. For the rest of this section, we will refer only to ship records, but all the remarks also apply to the convoy records.

We first thought of creating one track record per ship, or even putting this information in the ship file. However, due to current limitations of the Datacomputer, this file would have been extremely difficult to update. The only way to update such a record would have been to delete it entirely, and recreate it; moreover, a large amount of unused space would have been created in the file in this mode.

A second solution would have been to create one file per ship, but this would have taken too long to create, and would probably be very wasteful of space on secondary storage. However, this solution may be the right one to implement in the future, should the Datalanguage may be extended to admit the notion of file groups.

In the solution we chose, we attempted to limit the access time inefficiencies in two ways. First, the last known and next estimated positions were redundantly stored in the ship file, on the expectation that these are the positions most queries will want to find.

Second, we created some special year and month fields in addition to the complete ten-digit date field, and inverted them, so that a query may efficiently ask for a time-limited subset of the positions concerned with a specific ship.

The track file was created in a fixed length format for the same reasons as the ship file.

3. Embarked Units

The third file is EMBARKEDUNITFILE. It contains general information about embarked units, such as their name, their commanding officer, and the ship on which they are embarked. For aviation units, it also contains information about their permanent base, and their state of readiness.

This file was created in variable length format. However, it appears that the use of this format would create some incompatibilities with the other fixed length files, due to some limitations in the present Datacomputer implementation; we are therefore considering the recreation of this file in fixed length format.

4. Convoys

The fourth file is CONVOYFILE. It contains information about convoys, such as their title and international radio call sign, information about their origin and destination, and their last known position, as well as their next estimated position.

This file was created in fixed length format for the same reason as SHIPFILE.

5. Ship Class Static Characteristics

The fifth file is CLASSFILE. It provides static information about the various ship classes, such as their country, length, beam, fuel type, and for naval ships their armament.

The file was created in fixed length format.

6. Ports

The last file is PORTFILE. It is not really part of the database as NELC defined it, but it is useful in answering many queries about the data, and therefore was created in the same directory. It contains the ports' names, countries and geographic coordinates.

It was created in variable length format.

7. Philosophy Underlying Design Decisions

We did not attempt to specify an ideal physical format for some particular operational environment. The tradeoffs between efficiency of update versus efficiency of query, and of ease of error checking versus efficiency were not considered. We attempted to create a realistic database organized in a way similar to existing databases. The purpose was to give researchers a resource that could be considered as close to a typical situation as possible.

The following section gives the complete description of these six files in Datalanguage format. It includes in the comments an explanation on the semantic meaning of the fields, and in some cases, some typical examples of their values.

The next section gives the field values for all the records in the database, as it now stands.

D. File Access Management

1. General Description of Goals

To develop a complete facility for interrogating remote data bases, we must take the data base queries developed by the query-answering and data base access planning portions of the system and use them to access the appropriate files. The general solution to this problem involves the development of a facility for the management of

distributed data in a network environment, which is a substantial research topic in its own right.

In order to develop a functionally complete query answering capability at the earliest possible date, we have designed a more special-purpose facility that is oriented toward the particular near- to intermediate-term needs of the Command and Control Testbed. This facility will not be a general file manager; it will manage <access> to files. It will not deal with TENEX files in general; it will be specifically designed to deal with <Datacomputer> files. While we will use this facility for our own query-answering system, we will provide it with a clean, well-documented interface so that other researchers may use it as well. We call this facility the <file access manager>.

Subsequent subsections describe the full file access manager, and an initial, zeroth-order version that has been developed during the current period.

2. Specification of the File Access Manager

The file access manager will accept command strings from a user or from a calling program. The command strings will consist of the generic name of the file or files to be accessed, an optional indication of the priority of the request, and a set of Datalanguage to be used to access the named files. The file access manager will develop an access link between the calling program and the particular file or files in a particular directory on a particular Datacomputer. Then the Datalanguage will be used to retrieve information from that file.

The file access manager will be documented and made available as an independent capability in the Testbed environment. It will provide the following features:

a. Location independence

Access to a file will be specified by generic name. The location of the file (i.e. a particular directory on a particular Datacomputer) will be

determined by table lookup. If the primary location is inaccessible, a set of alternative locations, also determined by table lookup, will be tried. Optionally, appropriate messages will be returned to the calling program.

If more than one file is specified, copies of all of them will be assembled at one location.

b. Fail-softness

If no response from the remote Datacomputer is made within some specified amount of time (say, one minute), then the calling program will be notified and an attempt will be made to access the named file(s) at another location. This will prevent the system from hanging when a remote site crashes during command execution.

If no known location of a given file is accessible for retrieval, all accessible locations will be polled to determine if any have a copy of the named file. If any copies are found, the most recent one will be duplicated in a temporary location, and that duplicate will be used for retrieval. Optionally, appropriate messages will be returned to the calling program.

c. Retrieval priority

Requests for retrieval may be issued with an indicator of some degree of priority. (If no indication is given, a minimal priority will be assigned to the request.) The degree of priority will be communicated in Datalanguage to the remote Datacomputer. Some modification to the Datacomputer software will be necessary so that the Datacomputer may respond to such a priority request.

3. The First Step: A Robust INTERLISP-RDC Interface

As a first step toward the development of the file access manager, we have developed a robust and efficient interface between INTERLISP programs and the RDC XXX that manages communications between a

local terminal and the remote Datacomputer. The interface was designed to satisfy three criteria. First, it should be extensible to satisfy the requirements of the full file access manager. In particular, we had to plan for the case where several Datacomputers would be accessed. Secondly, it was considered important that INTERLISP and RDC not share the same address space, so that major addressing errors in either of these two programs would not have any effect on the other. Keeping the errors well localized is necessary to satisfy the goal of fail-softness. Finally, the interface had to be reasonably time-efficient. This precluded the use of files to transfer information between the two programs.

In the solution we implemented, we use a slightly modified version of RDC. This modified RDC runs in a fork which is a descendant of the LISP fork. This fork is totally independent and asynchronous from the LISP fork, so that it would be possible for LISP to continue running and perform background computations while a long request was being processed by the Datacomputer. In this design, it would be easy to make the RDC fork (or forks if several Datacomputers have to be accessed) be a sibling of the LISP fork rather than a descendant, or even be in a completely different job. The transfer of information between LISP and RDC is done via a pseudo-teletype: LISP sends the Datalanguage on the pseudo-teletype, and RDC sends the data back on the same pseudo-teletype. If required, the Datacomputer messages can be shown on the user's teletype, or be only logged on a file, but in no case are they sent to LISP. In essence, LISP sees the Datacomputer as a very friendly DBMS which never makes any error, and always sends some data in response to the commands.

This design has only required a very limited modification of RDC, and a reasonably small amount of LISP code. The result has been surprisingly friendly in its behavior: most of the time the interface behaved exactly as we expected it to behave. It now looks that it will be a relatively easy programming task to extend this interface to a full file access manager.

E. A Real-time Query Answering System

A major goal of our efforts in support of the Testbed is the development of a text understanding facility that will enable a user to query a data base in a rich subset of English. Our approach to this goal is to adapt and extend the SRI Speech Understanding System to process textual inputs related to this subject domain. Our work with the understanding system is described in Section .

As soon as research with a realistically large data base began, it became clear that having even a very limited natural language access to the data would both facilitate our own work and make the data more easily available to other interested parties. We desired to develop an experimental tool that could be used to acquire protocols of many users interacting with a data base. We wished this tool to run in real time, so that it could be placed in the Testbed environment and provide us with data about the kinds of questions that are asked. Therefore, in tandem with work on a sophisticated language system, limited resources were devoted to creating a simple facility for handling the more basic data queries.

This simple system does not understand a query in the sense that it does not not build an internal representatin of the meaning of the query. Rather, each query is parsed and then reponded to without ever determining its precise meaning. Our reasons for developing this alternative natural language interface are as follows:

* To develop a better understanding of the limitations of this approach--By stretching the capabilities of this system to the limit, we will learn in what ways the understanding approach provides the strongest benefits, and in what ways the simpler approach can be integrated with the understanding approach for the sake of efficiency.
* To provide an experimental tool for the acquisition of protocols of data base retrieval--A wide variety of users can interact with this system and by analyzing their queries we can build the appropriate language definition for the language understanding system.
* To provide a convenient vehicle for testing the data base access routines--Since this system is extremely efficient

and runs in real time, it is a natural tool to use for testing the file access manager and the data base access planner.

* To provide a useful capability for the Testbed--Even if the system accepts only a very limited subset of English queries, it can still be used by computer-naive individuals with a minimum of training. Thus, we can provide a useful feature for the testbed now while focusing on developing a much more powerful feature for the future.

The simple language facility was built using an SRI-developed package for building language interfaces, called "LIFER" ("Language Interface Facility with Elliptic and Recursive Features"). A simple applications-oriented system, LIFER emphasizes ease of use and flexibility. The system offers a range of capabilities that support both simplistic intefaces and language definitions of considerable complexity. This range of capabilities allows casual users to rapidly define workable interfaces while providing more advanced users the tools needed to produce more powerful and efficient systems. LIFER includes an automatic mechanism for handling certain classes of ellipical inputs.

The simple interface allows us to ask such questions as:

How long is the Lafayette?
When was the Whale built?
How many guns does the Kiev have?
What nuclear submarines have a submerged speed of
moe than 30 knots?
What cruisers did General Dynamics build?
What subs built in 1970 are longer than the Revenge?
Where is the fastest US carrier based?
How many Lafayettes are there?

Using the LIFER ellipsis feature, sequences like the following are also possible:

What is the speed of the Lafayette?
Of the Ethan Allen?
Displacement?
Length of the fastest Russian sub?
Slowest?

The current system permits access to a Datacomputer file of static characteristics of 741 ships. The system uses the LISP-RDC interface

described in Section . The system may be used to query any of the following attributes: anti-aircraft armament, anti-submarine armamaent, aircraft, beam, builder class, date of commission, complement, country, designation, draft, engines, flight deck width, full displacement, home port, hull number, date laid down, date launched, length, missiles, name, number of catapaults, number of guns, number in class, number of missile launchers, numnber of missiles, number of torpedo tubes, number of torpedoes, power, speed, standard displacement, submerged displacement (for subs), surface speed (for subs), and type.

The grammar for queries about this data file consists of about twenty-five top-level patterns, and an equal number of subpatterns. With a grammar of this size, the system requires less than two seconds of CPU time to parse a query. Since the Datacomputer can answer a query against this single file in less than twenty seconds of real time, the overall system is quite comfortable to use interactively.

Since the system is a simple one, it has clear limitations. It is very weak on handling pronouns. For example,the system always interprets "it" to mean the last reference to a ship or set of ships. This is appropriate for queries against the single ship file, but will not be extendible to a richer domain of discourse.

The grammar required to handle queries against the single file seems to be of a size that the system can handle easily. It remains to be seen if the grammar needed to handle queries against the full command ad conrol data base will be small enough for the system to cope with.

The current system builds specific data base queries for each kind of question that can be asked. When the system is converted to handle the full data base, a more general approach to data retrieval will have to be used.

F. Query Understanding on Textual Input

To provide comfortable access to information in the Testbed environment, we are in the process of developing a robust natural language understanding capability that will build upon work conducted by the SRI speech understanding project [4]. With far broader scope than the short-term system discussed above, this undertaking aims for direct translation into a representation of the meanings of queries and statements. The internal representation we are using is a partitioned sematic network [5]. The representation of the meanings of queries may be manipulated by the knowledge- based systems which will do problem solving and plan data base accesses. Our emphasis during this initial contract period has been to cut a complete, although narrow, vertical slice of the envisioned system. While the performance of this slice is well below the performance level of the short term system, it represents an advanced technology with growth potential for rapid broadening into a much more comprehensive system. Even in this preliminary state, this slice clearly demonstrates an ability to translate from English text into networks, to use network-encoded knowledge both in guiding translation and in directing calls on a remote data base, to convert information retrieved from a data base into network structures, to generate English outputs based on such retrieved structures, and to use network-encoded discourse histories in resolving anaphoric references and expanding elliptical inputs.

The speech system upon which our approach is based is documented elsewhere [6]. Briefly, control of the understanding process is resident in a language executive. This executive coordinates language analysis resources in the task of understanding inputs. The executive is guided in this task by a language definition. This definition, which is based on studies of dialogs between users and idealized information systems, provides rules for combining input fragments to form phrases and, ultimately, complete assertions, commands and queries. These composition rules may call for the invocation of various language processing specialists and reference information from a variety of

knowledge sources, including syntactic, semantic, and contextual information. In the Testbed environment, an important source of guidance is domain dependent knowledge which is encoded in a semantic network. Each composition rule indicates how to compute properties of the newly formed constituent from its component parts and how to determine the likelihood that the new constituent is a correct interpretation of a portion of the user's input. These likelihoods are used to coordinate the allocation of language understanding resources while the computed properties are used to divine the linguistic form and meaning of the input sequence. When a complete input has been recognized, the purport of the utterance is expressed formally as a network structure. This network structure is then given to systems which decide how to respond to the input.

1. Adapting the Speech Understanding System for Textual Input

To adapt the language understanding system for operation in the CC environment, a provision was added to drive the parser with text input. This conversion has been complicated by two factors. First, the parser, expecting to handle speech, builds numerous internal structures which are superfluous in processing perfect text. (It should be noted, however, that such structures may well be needed to deal with errorful input.) We have begun the job of removing unneeded structures to optimize the system for text, but some work in this area remains to be done. Second, acoustic processing was unable to reliably determine normal word boundaries and and thus would "hear" suffixed words as two (or more) separate entities. For example, "OWNS" may be heard as the verb-stem "OWN" followed by the suffix "-S," which signals a third person singular construction. For families of words such as (OWN, OWNED, OWNING, OWNS) or (SHIP, SHIPS, SHIP'S, SHIPS'), a text processor must either have a separate lexical entry for each word or some means of dissecting family members into a common stem followed by a particular suffix. Since large vocabularies are expected in the CC domain, a

lexical stripper has been implemented to determine the stem of any word. This eliminates the need to store multiple lexical items for variations on the same word. The stripper considers only what are called "productive" affixes -- those admitting algorithmic determination of the meaning of the original word given the root and affix(es), without recourse to special, "lexical" flags or markers. As an added bonus, it expands contractions (-N'T, -'RE, -'VE, -'LL, -'M, -'D, -'S) into their full forms, separates any appended punctuation, and converts the determiner "AN" to "A". This program began as an implementation of the flowchart presented by Winograd [7]; in the process of testing and evaluation, several bugs were excised and the algorithm was made more efficient. Eventually, its capabilities were significantly extended.

The stripper handles a complete range of regular verb endings (-S, -ING, and -ED), noun endings (-S, -'S, and -S'), adjective endings (-ER, -EST, and -LY), and ordinal number endings (-ST, -ND, -RD, and -TH), some of which involve minor spelling changes (for example, consonant doubling). In addition, it handles almost all of the -EN verb endings plus many irregular verb forms exhibiting internal vowel shifts, as well as several noun plural variants drawn from Latin or Greek (forexample, FORMULAE, QUANTA, THESES, SCHEMATA). If all other attempts fail to produce a root, the program checks for one of several negative prefixes (for example, NON-, UN-); if found, the prefix is removed and the remainder is once again checked for a suffix. In all cases, if the proposed root scores a lexical "hit", a test is performed to insure that the suffix is in fact appropriate to at least one sense of the root: if the test fails, the interpretation is rejected and others may be tried. (This strategy also serves to disambiguate roots with multiple senses, based on syntactic class.) If all attempts fail to disclose a root, the stripper calls the user-supplied function SPELLING-ERROR to take appropriate action.

An evaluation of this stripper indicates the added overhead to be on the order of .01 seconds per word.

Even with lexical stripping, the Testbed task domain will soon place such heavy lexical demands on the PDP-10 INTERLISP system that some new method for storing and retrieving lexical data will have to be found. A mechanism for maintaining an extensive lexicon in multiple subordinate forks is currently under development.

2. Current System Status

The domain of discourse of the text understanding System is defined by a data base of information about ships of the US, Soviet, and British fleets. There are 226 individual ships, falling into 76 classes and into two ways of categorizing type. For each ship, there is information of the following kinds:

Inputs can be formulated that relate to attributes of a particular individual ship or of ships meeting a certain description; to part-subpart relations between a ship and, for example, its crew; to set membership and kind relationships between various individuals and classes (such as "all ships", "Are all ships nukes?"). It is possible to specify an individual on the basis of its properties ("What country owns the Skate?", "What American destroyer has a speed of 33 knots?") or of the number of individuals meeting a given description ("How many ballistic missile subs are owned by the US?") Queries may be quantified to seek information over classes of individuals ("What is the speed of each American sub?") This information can be elicited by questions or commands ("Print the length of the Seahorse"), and statements about the data can be recognized. Previous utterances in a series of queries serve to provide a context, so that pronouns can be used, the referents of determined noun phrases can be identified, and incomplete utterances can be understood if the reference is clear ("What is the surface displacement of the Lafayette?", "The Ethan Allen?" "Submerged displacement?", "What is the speed of it?").

At this early stage, this vertical slice has many limitations which must be attacked in future work. Aside from the need to broaden

each layer in the vertical slice, certain imbalances exist among the current layers. Statements cannot yet be used to update the system's knowledge base, nor is their content checked against it for consistency. More importantly, more questions and commands can be understood correctly than can now be answered. That is, semantic processing may identify the meaning of a question, which would be translated appropriately into network structures, but the routines necessary to derive the answer from the data base may not be ready ("Who owns each sub?" involves universal quantification, and "What subs were built by General Dynamics?" currently returns only one answer). In discourse, elliptical quantifiers are not yet handled, and if there is no determiner a default determiner is assumed. Intrasentential pronominal reference is not yet handled, and responses to user queries are not yet included in the discourse history.

The following examples illustrate more systematically some of the kinds of utterances that the system can process. Those marked by "(T)" may be translated but not answered.

WH Questions

NP VP	Who has nuclear subs? What country owns the Skate?
NP BE NP	What is the speed of the Barb?
NP BE VP	Whose frigates were built by Avondale Shipyards? (T)
NP DO NP VP	Which cruiser does the US own?

How Many Questions

NP BE NP	How many subs are diesels? (T)
NP DO NP VP	How many subs does the US own? (T)
NP BE THERE	How many Lafayettes are there?

How + Adjective Questions

HOW ADJ BE NP	How long is the Tullibee?

Yes/No Questions

BE NP NP	Is the Whale a Russian cruiser?
	Are all Lafayettes nukes? (T)
	Is the speed of the Lafayette more than 30 knots?
BE NP VP	Were any carriers built by New York Naval Shipyard?
DO NP VP NP	Does the US have more than 5 destroyers? (T)
DO NP VP	Does Russia own frigates?
BE THERE NP	Are there any diesel subs?
	Are there more than 5 cruisers? (T)

Imperatives

VP	List all Russian subs. (T)
	Print the length of the Ark Royal.
	Name a builder of Skates. (T)
	Give the speed of each sub. (T)

Statements

NP BE NP	The complement of the Skate is 75 men. (T)
	Vickers Armstrongs Limited is the builder of the Courageous. (T)
NP VP	Vickers Armstrongs Limited built the Courageous. (T)
	Russia owns the Sevastopol. (T)
NP BE VP	The Courageous was built by Vickers Armstrongs Limited. (T)

Noun phrases in utterances can be a variety of types. We have concentrated on those relevant for the data base having WH determiners (what, which, whose), quantifiers (all, any, both, each, either, every, neither, no, none, some), partitive expressions (containing "of"), expressions with numbers (from 1 through the millions) and units (tons, feet, knots), and comparisons involving numbers. There are about 90 different kinds of basic noun phrases, to which may be added recursively "of NP" expressions and some classes of prepositional phrases. Examples of acceptable noun phrases are: which US submarines, all British submarines, every builder, a displacement of seven thousand tons, more than six feet, three, the guided missile cruiser.

In the course of a dialog about the data base, it is not necessary to use complete sentences when the reference to the prior utterance is clear. Pronouns and definite determiners will identify relevant data from previous utterances, and elliptical expressions will be filled in, as indicated:

Definite Noun Phrases (non-pronominal)--resolved in local context ... the Darter ... (a submarine) ... the Saratoga ... (an aircraft carrier)

Does the submarine belong to the US? (uses Darter)

Do both ships belong to the US? (uses Darter and Saratoga) Ellipsis--expanded in context of immediately preceding utterances

Does the United States have a diesel carrier?

A nuclear carrier?

What is the surface displacement of the Alexander Hamilton?

The Barbel?

Submerged displacement? Pronominal References--resolved in terms of immediately preceding utterance

How many diesel carriers does the US have?

How many nuclear carriers do they have? (they = US)

Is the Woodrow Wilson a nuclear ship?

Is it a nuclear sub? (it = Woodrow Wilson)

There are over 600 entries in the current lexicon, plus plurals, past and past participle forms, and suffixes. The result is over 900 base forms.

G. Recent Developments in Network-Data Base Linkage

To allow a complete vertical slice of the Testbed system envisioned for 18 months hence to be implemented during the contract period just ending, we have cooperated with the SRI speech understanding project in a joint effort to couple a semantic network to a relational data base. Although the flow of information between net and relational file is, in this pilot system, still meager, the interconnection is itself a

milestone in the development of intelligent access to data bases which we believe to be of fundamental importance.

In our previous work in natural language, the burden of producing responses to user queries has fallen primarily on a system which retrieves information from a semantic network. This system, called the "network matcher," takes as its input a net fragment, containing vacant "slots," which is to be matched against structures in the semantic network encoding the system's general world knowledge. The matcher's job is to find areas within the knowledge net which resemble the input net (or "query net") and to fill the vacant slots of the query net. Typically, the query net with which the matcher is called is the network produced by the natural language translation process.

To perform its job, the matcher follows a constraint satisfaction algorithm that first attempts to find a structure in the knowledge net which matches the input directly. If this direct matching process fails, the matcher attempts to derive new facts whose network representations might match the structure of the query. To derive new structures, the matcher may use its knowledge about the transitivity of certain primitive binary relationships which are expressed in the network. If this also fails, the matcher may use theorems to derive new information. Such theorems are themselves encoded through the partitioned network formalism.

The new link between nets and relational data bases builds upon the network's ability to represent theorems and the matcher's ability to manipulate them. The approach is simply to include theorems within the system which indicate that needed information of various types may be obtained by pulsing the data base in specified ways.

Figure 1 shows a theorem for associating a ship with its builder and vice versa. The encoding of the theorem uses two overlapping partition spaces. In the figure, the upper space is used to represent the theorem's ANTECEDENT and the lower space the CONSEQUENT. This theorem may be used to produce new structures matching the form of the CONSEQUENT.

The CONSEQUENT structure represents an abstracted building event B, an element (e arc) of the set of all building events, S.BUILD. Building event B has two arguments or deep cases: an object (obj) that is built (I), and an agent (agt) who builds the object (P). Because B, I and P lie within a consequent space, they represent variables and hence encode no building event in particular.

If needed, the matcher may assign values to I, B and P and produce a new structure matching the consequent space. But to do this, the matcher must first find matches for the structures in the ANTECEDENT space. Further, the bindings of both I and P must be the same in both spaces.

To match the ANTECEDENT space, I, the object built, must be an element (e arc) of some set C which is an element of the set of ship classes, SHIP.CLASSES. Hence, this is a theorem about builders of ships. To find a match for R, the matcher must find an element of the set S.FILE.RECORDS. This is a special set whose members are not usually recorded in the semantic net but which are kept on a relational file. In other words, R is the network representation of a record on some file. The particular file upon which R must appear (if it exists) is indicated by the "table" arc. For this example, R must be a record on file TABLE4. The other arcs leaving R (the "class" and "builder") indicate that TABLE4's records have an entry for CLASS and BUILDER.The force of the theorem may thus be stated as follows:

> If there is a record on TABLE4 with C as the CLASS entry and P as the BUILDER entry and if C is a ship class with element I, then P built I.

This theorem may be used to provide network answers to such questions as:

> Did General Dynamics build the Abraham Lincoln?
>
> Who built the Abraham Lincoln?
>
> What ships were built by General Dynamics?

Theorem chaining is possible. For example, to answer

Who built the Abraham Lincoln?

it may be necessary to use a theorem to determine to which class of ships the Abraham Lincoln belongs.

REFERENCES

1. DECsYSTEM 10, "Data base management system programmer procedures manual," DEC-10-APPMA-B-D, 2d ed.

2. Data Base Task Group of CODASYL Programming Language Committee Report, available from ACM, April 1971.

3. W. Zook et al. "INGRES Reference Manual-Version 6," Memorandum No. ERL-M579, Electronics Research Lab, UC Berkeley, April 1976.

4. D. E. Walker et al., "Speech Understanding Research," Annual Report, Project 3804, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, Ca., June 1975.

5. G. G. Hendrix, "Expanding the Utility of Semantic Networks Through Partitioning," Proc. Fourth International Joint Conference on Artificial Intelligence, Tbilisi USSR, pp. 115-121.

6. G. G. Hendrix, "Semantic Processing for Speech Understanding," AJCL Microfiche 34, 1975, 34-48.

7. T. Winograd, Procedures as a Representation for Data in a Computer Program for Understanding Natural Language. Report MAC-TR-84, Project MAC, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1971. [Published as Understanding Natural Language, Academic Press, New York, 1972.]

V INTERACTIVE AIDS FOR CARTOGRAPHY AND PHOTO INTERPRETATION

by

Harry G. Barrow, Thomas D. Garvey, Jan Kremers,

J. Martin Tenenbaum, and Helen C. Wolf

A. Introduction

1. Overview

This report covers the six-month period October 1975 to April 1976. In this period, the application areas of ARPA-supported Machine Vision work at SRI were changed to Cartography and Photointerpretation. This change entailed general familiarization with the new domains, exploration of their current practices and uses, and determination of outstanding problems. In addition, some preliminary tool-building and experimentation have been performed with a view to determining feasibility of various AI approaches to the identified problems. The work of this period resulted in the production and submission to ARPA of a proposal for research into Interactive Aids for Cartography and Photointerpretation. This report will not reiterate in detail the content of the proposal, but will refer the reader to it for further information [1].*

2. Selection of Domain

There were three essential criteria in selecting a domain for Image Understanding research; it must be of importance to the Department of Defense, it must present central issues of fundamental scientific importance for study, and it must present problems that are tractable.

Information in the form of aerial photographs is of prime military importance, for both strategic and tactical purposes. Many

----

* References are given at the end of this chapter.

thousands of photographs are taken each day and used for making maps, and for obtaining intelligence information. Current cartographic and photointerpretive techniques require substantial use of data processing, but computers are not yet used directly for processing the photographic data. There are still labor-intensive bottlenecks in the conversion of raw pictorial information to a symbolic form. In map-making, for example, cartographic features such as roads or lakes are traced by hand, while in photo interpretation, even basic tasks like counting or measuring are perormed manually. It is therefore evident that there is considerable scope for automation and consequent increase in quality and quantity of throughput. So far, however, these tasks have resisted automation, primarily because a task that appears simple, like road tracing, involves application of considerable amounts of knowledge. Tricky cases that would defeat an elementary approach occur very often. The approaches of Artificial Intelligence, and Machine Vision in particular, are aimed directly at the embodiment of diverse sources of knowledge in computer programs which are then able to display useful expertise in selected tasks. The application of AI to cartography and photointerpretation is therefore natural and appropriate, and promises to be fruitful in the forseeable future. We have, in consequence, selected the twin domains of Cartography and Photointerpretation for our next program of research in Machine Vision.

At first sight, Cartography and Photointerpretation may appear to be totally different fields. Cartography is concerned with locating the permanent features of an area with great accuracy, while Photointerpretation is concerned with dynamic features and events. In fact, they are simply extremes of a continuum, and overlap considerably. In order these two fields map features, one must first interpret the photograph, and in interpreting a photograph, one implicitly constructs a map. We therefore deem it inappropriate to draw an artificial distinction, and believe that there is much to be gained, in terms of both expediency and scientific interest, by taking a unified view of Cartography and Photointerpretation.

3. Scientific Objectives

The crucial issue in Image Understanding is the role of knowledge. Many important questions are still very much open. What domain-independent knowledge should be built into the lower levels of a visual system? What domain-specific knowledge is necessary for achieving specific goals? How should that knowledge be deployed, invoked, and applied? How can many diverse sources of knowledge be coordinated? The only irrefutable fact is that knowledge, both general and specific, is essential to image understanding.

The work that we have embarked upon in the present period includes an in-depth study of a particularly potent embodiment of knowledge which is of both theoretical and practical importance, namely a map. A map, which can be viewed as a data structure that preserves three-dimensional geometrical and topological properties, can be of very great assistance in interpreting a photograph.

The geometrical relationships between maps, photographs, and the real world are well understood, and it is possible to establish an exact correspondence between points in the map and points in an image. Thus, the map can be used to predict which features should appear where in the photo, thereby eliminating much general-purpose searching and processing. It appears that using the map to guide the processing and interpretation of an image may yield considerable gains in efficiency and robustness, as well as providing a coherent, modular structure for the whole system.

Within this framework, a variety of different types and levels of knowledge may be unified. To give a specific example, advice provided interactively by the user in the form of approximate manual tracings of a road may be regarded as additional information which may be stored and manipulated in map form, and which may be used along with general knowledge of the characteristic behavior of roads in locating the road accurately in the image.

The scientific objectives of the current work are thus the elucidation of what information should be stored in the map data base, how it should be represented, and how it may be employed flexibly in image understanding.

4. Application Objectives

In the domains of Cartography and Photointerpretation, we are focussing our attention on those bottleneck areas upon which we expect to be able to make significant impact within a reasonable time scale. The cartographic tracing problems are currently under study, since these also underlie many photointerpretation tasks. Gradually, the emphasis will shift toward photointerpretation.

We recognize that it is not possible within the current state of the art of machine vision to fully replace human abilities. We therefore adopt as one important objective the development of a system that can accept advice supplied interactively by a human user and collaborate with him in an image understanding task.

In summary, we aim towards the development of a collaborative aid for a cartographer or photointerpreter that can employ information provided in the form of a map, or supplied interactively by the user.

5. Summary of Work Carried Out

The work of the current six-month period has been a broad exploration to determine techniques and approaches currently in use in production cartography and photointerpretation. The work has included the design of and experimentation with a basic integrated system.

The main accomplishments of the six-month period are as follows:

* We have equipped ourselves with a range of documents covering civil and military mapping and photointerpretation. We now have a reference library that should prove valuable in our current and future research.
* We have established contact and visited a number of

centers at which civil and military map-making and photointerpretation are performed, and research centers at which approaches and techniques for the future are being developed.

* We have determined bottleneck tasks in Cartography and Photointerpretation as currently practiced. We have identified the common requirements, and examined their susceptibilities to Machine Vision techniques.
* We have acquired a realistic imagery data base, representative of the important tasks and central issues.
* We are currently in the process of designing and implementing a basic interactive cartography and photointerpretation system for supporting our research work.
* We are already using the basic system in some initial experiments in map-guided tracing.
* We have written and submitted to ARPA a proposal for further research on Interactive Aids for Cartography and Photointerpretation.

The proposal contains a detailed program of research, with a description of a proposed interactive system. Therefore, we shall not dwell further on future research, but confine ourselves in the rest of this report to work completed in the present six-month period.

The next section reports in detail on the studies we have so far made into the current approaches to cartography and photointerpretation. We shall identify certain common requirements, their susceptibilities to AI techniques, and the key obstacles that must be overcome.

The succeeding section describes the experimental work performed so far, including an examination of the problem structure, acquisition of suitable experimental data, the design of an experimental system, and some initial experiments in interactive map-making.

The final section gives general conclusions concerning the research and applications upon which we have embarked, including observations relating to the guidance of the proposed future work.

B. Survey of Applications and Requirements

This section summarizes the current status of Cartography and Photointerpretation, highlighting the bottleneck areas on which our project is now focused. Our information was gathered from the reference material listed in Appendix I and from visits to the facilities listed in Appendix II.

1. Cartography

a. Present Approaches

Despite a large body of mechanical techniques, the production of maps from aerial photos is still primarily labor-intensive. The most automated systems now in existence are the SACARTS (Semi-Automatic CARTographic System) developed for DMA-TC by USAETL and the LIS (Lineal Input System) developed for DMA-AC by RADC. A schematic of the SACARTS system is shown in Figure 1.

SACARTS includes facilities for automating all stages of map production, including compilation, editing, and reproduction. Our discussions will concentrate on the *compilation* and *editing* phases, where AI techniques can contribute most directly. Compilation consists of the acquisition in digital form of the cartographic data to be mapped. The two major types of data are elevation profiles and boundary coordinates of planimetry (lineal features, such as roads, rivers, and coastlines). Editing is concerned with verification of the internal consistency and accuracy of the information, and its correction where necessary.

The first step in compilation is performed by an automated stereo plotter, known as UNAMACE (UNiversal Automatic Map Compilation Equipment). UNAMACE takes an overlapping pair of unrectified aerial photographs and produces both an orthophoto (a rectified photograph in which all terrain points are depicted in their correct map positions) and a set of elevation profiles.

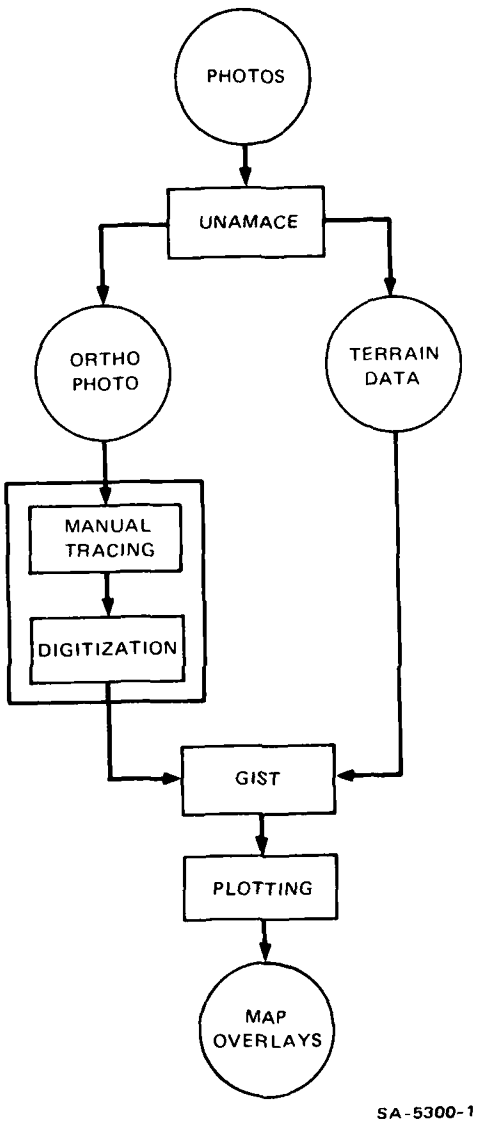


FIGURE 1 SCHEMATIC OF SACARTS OPERATION

The elevation data are smoothed by software and transformed into contours, for subsequent merging with planimetry extracted from the orthophoto.

The extraction of planimetry has always been the most labor-intensive and time-consuming step in map compilation. In SACARTS, each class of planimetry is manually traced on a separate sheet of frosted acetate that is overlaid on the orthophoto. (Planimetry can also be compiled by tracing features on a pre-existing map.) The acetate sheets are then transferred to a digitizing table where the features are retraced to produce a digital tape containing their digital coordinates. This operation concludes the compilation phase.

The digital elevation contours and planimetric data on magnetic tape are now transferred to a Univac 1108 where they are manipulated by a series of editing programs, known collectively as GISTS (Graphic Improvement Software Transformation System).

The data are first output on a high speed plotter for manual verification. Boundary and contour imperfections are currently corrected off line by retracing on the digitizing table and creating an update tape. However, an on line graphics editing system (DIODE) will soon be available.

From this point, processing becomes more automatic. Linear features of definite width, such as roads, are thinned down to a center line and smoothed. Precise intersections of features on a single overlay are computed. Selected features are checked for positional correctness against independently entered ground control coordinates. Features on different overlays are then checked for consistency. For example, the system checks whether any buildings are positioned on top of roads. Inconsistencies are resolved automatically where possible, and manually otherwise. The system is currently capable of resolving a road/building conflict by displacing the building and orienting it in the direction of the road. The system will soon be able to check for other types of local consistency, such as whether roads follow contours

and streams cross contours, and whether bridges are indicated at road/river intersections.

b. Limitations and Needs

SACARTS, though still not fully operational, promises significant savings in time and manpower over conventional map making technology, plus a reduction in the opportunities for human error. These advantages are realized primarily in the automation of contour extraction and in the manipulation of extracted data into a form suitable for a finished map. The system still leans heavily on human inputs for the extraction of planimetry, for the extraction of contours in difficult terrain, and for error correction during editing. The following sections discuss these bottlenecks, the currently contemplated solutions, and the possible contributions of AI research.

Extraction of Planimetry

The major remaining bottleneck is the tedious manual tracing that occurs in both the extraction and the subsequent digitization of planimetry. The amount of labor required in these steps, both in hours and in dollars, is phenomenal. Typically, it takes about 75 hours to trace all the overlays (roads, streams, drainages, buildings, and the like) for a small scale (7.5′) topographic map of a rural area. In a heavily urbanized area, it can take as much as 500 hours. This tracing is performed by GS-9 level personnel, whose annual salary is about $15,000. Almost 400 of these people are employed at DMA-TC alone, and constitute at least a quarter of all personnel involved in map production generally. Tracing operations are by far the most time-consuming steps in map compilation and make it uneconomical to keep maps up to date. Clearly, automating the extraction of planimetry is a prime application for image understanding.

Several projects are currently underway at ETL, aimed at alleviating this bottleneck. The Autocartography group, under Howard Carr, which originally developed SACARTS, is now trying to eliminate the

manual retracing required for digitization. An acetate sheet containing pencilled tracings of a single feature (such as roads) is placed on a high resolution drum scanner and digitized in a raster format. The digitized data are thresholded to extract the lines. Each line is then thinned down to a center line and smoothed to eliminate gaps. Finally, interconnections of the lines are computed to recover the road network. Some of theses features are found in the competing LIS system developed at RADC.

Preliminary tests verify the feasibility of this approach but at the same time, have underlined the practical issues that arise in processing high resolution cartographic data. Processing times of about one and a half hours per overlay on a CDC 6400 have prompted examination of more appropriate computer architecture, such as the Goodyear STARAN. Speedups on the order of 25 times have already been observed for selected operations.

The Computer Sciences group, under Larry Gambino, is setting up an interactive image processing facility that will be used to study the much harder problem of extracting planimetry directly from photographs. The group is also investigating a variety of image enhancement techniques that should facilitate extraction by both man and machine.

The Technology Development branch, under Bernie Scheps, has developed an interactive image processing system based primarily on analog techniques. Density slices from up to four bands of imagery can be logically combined to produce a binary overlay. Although density slicing is a very limited form of feature extraction, with multi-spectral imagery, it is frequently possible to find thresholds that will extract most instances of a given feature, say roads, in a particular image. The real time nature of analog processing facilitates the empirical selection of suitable thresholds; the operator turns potentiometers while observing the displayed overlay, and stops when the best one is achieved. The resulting overlay could be digitally

processed in a manner similar to raster scanned pencil tracings to obtain an approximate road network.

None of the research to date has attacked the problem of tracing planimetry in black and white aerial imagery. This process, in our opinion, is too difficult to automate completely at this time. We are therefore advocating an interactive approach wherein features traced crudely at free hand speeds are used to guide the automatic extraction of precise boundaries. In updating old maps, the map itself is used to guide the tracing of pre-existing features. The man can interactively refine the machine's output if necessary, eliminating the subsequent need for an explicit editing step.

The same techniques can also be employed in updating old maps from new imagery. Here, the old map itself is used to guide the tracing of preexisting features.

Elevation Contour Extraction

The UNAMACE, and all other automatic stereoplotters produced to date, use local area correlation to extract disparity information from a stereo pair of imagery. Thus they do not function reliably in steep terrain, where appearances can vary significantly with slight changes of viewpoint, or in flat featureless terrain, such as open water. They are also unable to recognize and adjust to planimetry. Another problem is the strict demands made on the quality of photography used in automatic stereocompilation. Currently, about 40% of the images are rejected because the equipment will not correlate continuously. Some of these images can be handled after reprocessing in the photo lab. Because of all these difficulties, a human's judgement and global viewpoint are needed to correct errors and to rescue the machine when it gets lost.

It should be possible to incorporate some of the needed judgement in a smart correlator based on AI techniques. For example, the knowledge that contours must close and cannot cross can be used to

constrain image correspondence in featureless areas. It should also be possible to combine the extraction of contours and planimetry so that the depth disparity can be used as a feature in extracting planimetry, and the resulting planimetry can constrain the placement of contours.

Editing

Improving the sophistication of automatic map evaluation and error correction is the subject of ongoing research at ETL. So far, attention has focused on the detection of local inconsistencies in the map. The automatic correction of such errors is, in general, a much harder task, which can require knowledge of the overall landform, road networks, and the like. Additional knowledge about the relations of roads, rivers, contours, overpasses, and other features, and about their corresponding appearance in images, is needed to ensure that a map conforms to all the standards set forth in [2]. As the level and the global scale required for error detection and correction increase, the need for AI techniques becomes more apparent.

Map Using

Cartographic data bases of the type produced by systems like SACARTS can be used for many purposes other than just for printing new topographic maps. ETL is currently investigating a variety of such applications. They include the generation of special-purpose thematic maps, the efficient updating of previously compiled maps, and a variety of military geographic intelligence applications, such as computing the cross-country travel times of a vehicle or the construction time for an airfield. The latter capabilities could be combined with AI problem solving techniques to augment manual decision making in many tasks. Some examples are: finding a best route between two destinations, subject to constraints; determining the best location for an airfield; balancing such factors as tactical surprise against construction time; and determining regions of critical terrain.

2. Photointerpretation

a. Present Approaches

To learn about photointerpretation as currently practiced, we visited the 544th reconnaissance wing of the Strategic Air Command at Omaha. There we received an unclassified briefing on PACER, an operational computerized information handling facility for photointerpretation and analysis.

The PACER system, shown schematically in Figure 2, consists of a number of photointerpretation stations and analysis stations, all on line to two Honeywell 6080 computers. The computers maintain a multi-source data base of intelligence information that serves as the principal means of communication between the photointerpreters and the analysts. The data base contains known target descriptions and related intelligence, expressed by both formal descriptors and free text elaboration. The data base also contains notes from analysts requesting further interpretation or monitoring of selected targets. There is a limited amount of on-line cartography, but no on-line imagery.

The analyst stations are conventional CRT text terminals. Analysts enter their photointerpretation requests which are communicated through the data base to the appropriate PI stations assigned to the geographic areas in question. Interpretations made by the PI are then entered in the data base where subsequently the analyst can examine them on line.

Each PI station has three components: a light table, a film chip viewer, and a Bunker-Ramo BR-90 display. The light table is equipped with two film transports and a conventional zoom microstereoscope. The chip viewer is augmented by a mechanical storage and retrieval facility. The display is a graphics console with light gun. Slides can also be rear projected on the display surface.

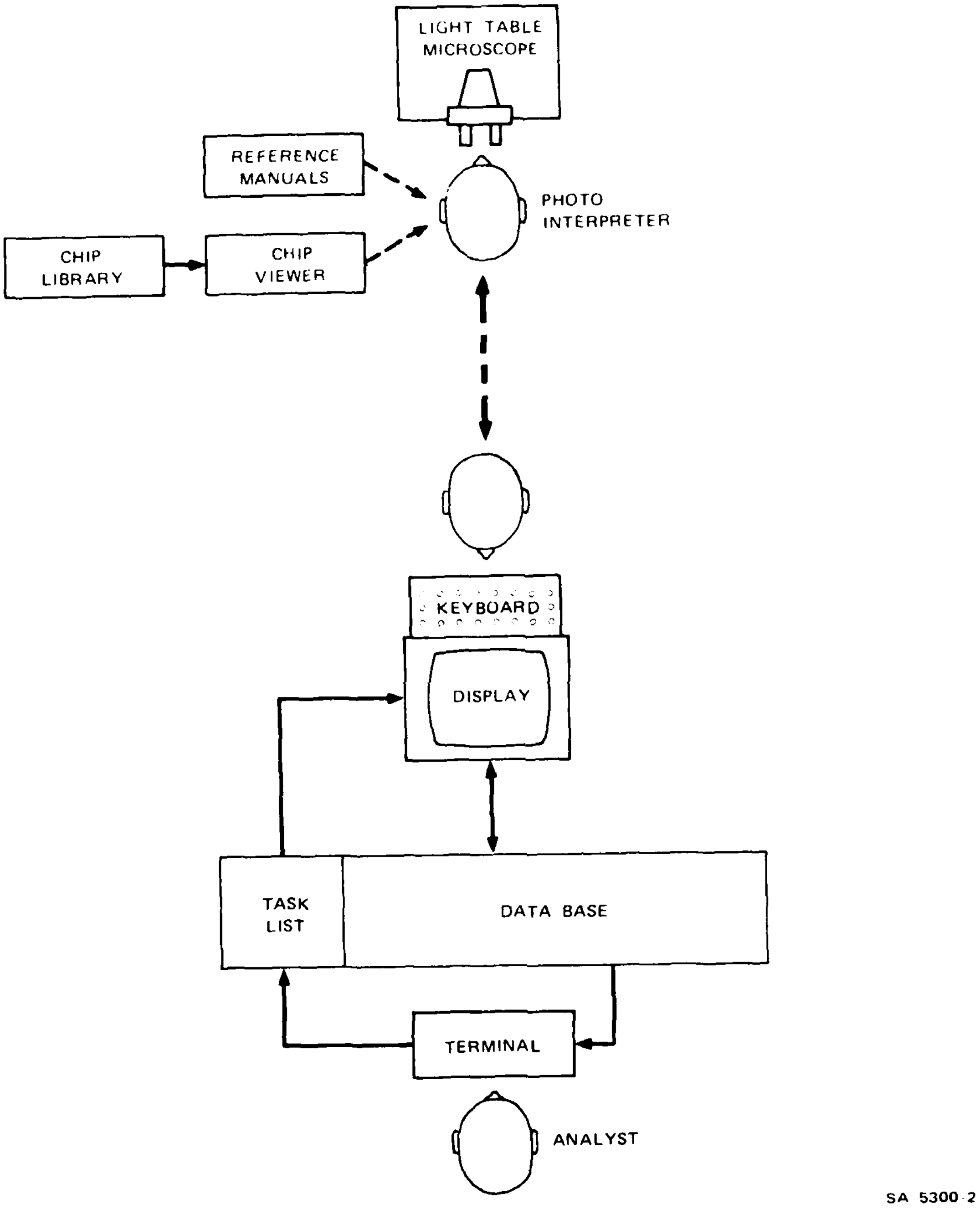


FIGURE 2 SCHEMATIC OF THE PACER SYSTEM

A pair of PIs are assigned to each station. One views the imagery on the light table, while the other interrogates the data base and retrieves chips as needed. Both are specialists in a particular geographic area and they switch roles halfway through their shift.

PIs work with PACER in two distinct modes, a priority mode, concerned with monitoring high interest areas in support of a particular mission, and a more systematic mode, concerned with detecting, identifying, locating, and reporting in detail on installations and activities whose existence was not previously known. In priority mode, the targets selected for monitoring are flagged on the display with respect to a rear projected map. This location is verbally relayed to the interpreter at the light table along with the analyst's instructions on what to look for. This active interpreter must search through through rolls of film to find frames containing the designated target. He then responds to the analyst's request (often with a terse phrase such as "no change"). The response is entered into PACER by the other interpreter at the console who is, in effect, serving as a natural language interface. In the second mode, the interpreter at the light table searches systematically for new areas of interest in each frame. When a target is encountered, the interpreter verbally describes some distinguishing features with which his partner can determine whether an up-to-date description already exists in PACER. Target location can be designated on the map, using the light gun, and the computer will transform the display coordinates to obtain the actual map coordinates. Additional features may be needed to distinguish among similar targets clustered in an area. If PACER has such a target, it will output a description which can then be checked against the image for verification. The resulting dialog leads to a new or updated target description.

b. Limitations and Needs

PACER has gone a long way toward facilitating the timely exchange of intelligence information among PIs and analysts, and it is well regarded as having improved the overall effectiveness of SAC's reconnaissance operations. Nevertheless, its performance is still limited by the fact that all actual operations on imagery are performed off line and completely manually.

It is impractical to automate completely human perceptual and decision making skills. However, there appear to be many interactive aids based on Artificial Intelligence technology that could improve the speed, cost, and reliability of existing PACER capabilities. We can describe briefly some obvious opportunities for aiding image archiving, image enhancement, and graphical communication, as well as pure image understanding.

Archiving

Photointerpretation requires a substantial amount of comparison with previous imagery on film chips. There is thus the need for cataloging and storing images in a way that permits efficient retrieval of those relevant to some area or operation. The relevance of an image is not simply defined by the area it covers, but may involve the interpretation of what it depicts. For example, if one desires to determine where a ship unloaded her deck cargo, it is necessary to find out whether there is any coverage of the actual unloading, and failing that, to retrieve all recent photos of the ship at sea to determine when the cargo disappeared.

Images should thus be accessible by content as well as location and time, and if possible, should be available for on line viewing and processing. The current ARPA-sponsored work on very large memories and data bases may find application here.

Enhancement

A great variety of enhancement techniques have been experimentally investigated, ranging from false color displays to mathematically sophisticated filtering. Appropriately used, these techniques can sometimes trivialize the detection of targets that were indistinct in the raw image. In practice, enhancement techniques are seldom used, except for special studies of high priority targets. One reason, according to a CIA source, is that the average photointerpreter lacks the background needed for selecting appropriate enhancements for particular tasks. This shortcoming suggests the need for a knowledge-based system to serve as an enhancement expert. Given a description (or pictorial examples) of the desired target and the expected background, such a system would select the appropriate enhancement. With on-line imagery and electronic viewing, the enhancement can be optimized for each area of the image.

Graphical Communication

The person manning the PACER terminal is serving essentially as a natural language interface between the person doing the actual photointerpretation and the computerized data base. This interface is both wasteful of skilled manpower and awkward because it forces the verbal communication of concepts that are primarily visual. Consider as an alternative, a single user system with both on-line cartography and imagery. Image and map can be superimposed optically or electronically and brought into correspondence by manually designating a few corresponding points in each. The system could then task the PI by encircling the target to be monitored directly in the image. New target detection could be simplified by encircling all previously known ones. New finds could then be reported by pointing at them with the light pen and having the computer extract directly the geographic coordinates and possibly other descriptive parameters (see below). The PI might also enter a verbal description in a restricted vocabulary, using one of the commercially available speech input devices (such as Threshold Technology's VIP).

Image Understanding

Aids for image processing tasks such as counting, measuring, and change detection offer potentially the highest payoff, but also involve the most technological risk. Image processing research, supported primarily by the military, has been underway for almost 20 years. Almost universally, the results have been lacking either generality or reliability. The reason, we believe, is because the unreliable techniques did not use adequate knowledge of the domain and the nongeneral techniques were inflexible in the use of what knowledge they had. ARPA's Image Understanding program was indeed founded on the belief that knowledge, appropriately used, could dramatically simplify many image processing tasks of practical importance.

The PACER data base already contains a substantial amount of information about the location and appearance of targets. This information provides powerful constraints on where to look in the imagery and what to look for. Here are examples of how such knowledge might be exploited in two classes of PI tasks.

Change Monitoring

The conventional approach to automatic change detection entails the warping of intensity normalized images into geometric correspondence, and the subsequent subtraction and thresholding of corresponding pixel intensities [3]. Such an approach is both computationally expensive and limited in effectiveness to images taken with the same sensor from approximately the same viewpoint, at the same season and time of day. The approach also is incapable of distinguishing between change that is significant and change that is insignificant in a military context. To paraphrase one contact "Change detection is easy in vertical photographs taken at noon from the same position and orientation. The challenge comes in dealing with low angle obliques taken from different positions with low sun angles."

The general change detection problem is admittedly beyond the state of the art. However, automating the more constrained problem of change monitoring, that is, checking a particular location for changes of a particular type, may be feasible. We have concluded that the use of knowledge-based search techniques [4] are appropriate to locate precisely the specific targets tasked for monitoring in each frame of imagery. A search for significant change can then proceed in a verification mode, guided by the previous target description and independent knowledge of how changes in viewing conditions transform appearances. Indeed, simple techniques like subtraction can often be used quite effectively in suitably constrained contexts. For example, it should be very easy to determine reliably whether a given ship is still in port, once the pier at which it was previously docked has been located.

Counting and Measuring

Counting tasks, such as determining the number of box cars in a railyard or oil wells in an oil field, and measuring tasks, such as determining exact runway lengths and orientations or the capacity of a reservoir, are among the more tedious and time-consuming of a photointerpreter's duties. Conventional mechanical aids now used in practice include sampling grids and image marking devices for counting, rulers, proportional dividers, planimeters, and the like for measuring. As a consequence, the counting and measuring required in reporting, for example, a new airfield and its associated equipment can take up to 3 hours.

Some work is underway at RADC and similar establishments to alleviate this condition. A developmental system, known as Compass Preview, intended as a successor to PACER, features several interactive measurement aids. For example, a section of an image can be moved across a cursor to compute true ground lengths independent of the current optical magnification. At a research level, a variety of analog and digital electronic systems have been built that detect target areas

(by density slicing, template matching, or feature extraction and classification) and then perform simple mensuration, such as counting the number of distinct connected components, or determining their areas and perimeters. Such techniques have proved quite effective in remote sensing applications, such as estimating crop acreage, especially when the classification criterion or slicing threshold is chosen interactively for particular images. However, military targets are not usually as homogeneous as crops, and it is unlikely that any single classification criterion or threshold will suffice to extract all targets (or all of a single, extended target).

We have several possibilities in mind for applying image understanding research to problems of counting and measuring. As with change monitoring, knowing where to look can substantially simplify the processing needed to discriminate relevant targets. For example, by first delimiting the runway area (either interactively or automatically with the aid of a map), it may indeed be possible to extract airplanes by some relatively trivial operation such as thresholding. Similarly, if the exact location of the rail lines in the image is known, box car counting becomes a one-dimensional template matching problem.

Having delineated an area of interest, the PI might be able to use interactive scene analysis techniques like those developed at SRI [5] to rapidly develop a target finding strategy that takes maximal advantage of that specific context.

3. Research Requirements

Cartography and Photointerpretation have a sufficient number of techniques and requirements in common to suggest a unified technical approach for their automation. Three components appear fundamental:

(1) A data base for storing map information.

(2) Techniques for using the knowledge contained in a map to guide image analysis.

(3) Techniques for establishing correspondence between an image and a map.

Maps play primary roles in photo interpretation and cartography, both as formats for output and as sources of knowledge for guiding image analysis. Maps will be represented by a data structure that is indexed by geography and content. This data base will contain dynamic intelligence information in addition to static cartographic and cultural features. It will also contain auxiliary knowledge needed for image understanding, such as the pictorial attributes of objects.

Techniques will be developed for using map knowledge to guide image analysis. For cartography, it is necessary to trace linear features, such as roads, rivers, railways , and coastlines, using as a guide an approximate trace that has been manually entered. Similar techniques can be used for verifying the existence of features whose presence is indicated in a pre-existing map, for such purposes as map updating or change monitoring.

The techniques for map guided image analysis presume a reliable means of establishing geometric correspondence between map and image coordinates. A general transformation can be determined, given sufficient pairs of corresponding points in the image and map. Initially, these pairs will be designated interactively. Eventually, they will be found automatically by using crude map/image correspondence based on navigational data to constrain the search for known landmarks.

The three components outlined above will facilitate development of special-purpose counting, mensuration, and change monitoring aids for photointerpreters, which rely on a map/image correspondence to constrain where to look and what to look for. The aids capabilities are described at length in the proposal.

C. Experimental Studies

1. Domain

It was decided that the task domain for the initial experimental studies should be cartography, because the ability to construct and augment maps seems an essential requirement for advanced

photointerpretation. In addition, we already had some experience with applying AI techniques to cartographic problems [6].

The approach adopted was therefore to construct a simplified system for interactive map-making and updating, which may later also support map-guided interpretation.

2. Problem Structure

The paradigm task addressed runs as follows:

> A new image of an area of interest is digitized and displayed. The user indicates a few features of known location by pointing at them and giving their coordinates. The system then computes the coordinate transform between picture and world, enabling it to display a map of the area (if one already exists) accurately superimposed on the photo. The user can make additions or modifications to the map by tracing on the photograph: the system can use crude manual tracing or an existing map to locate and trace accurately features in the picture, and hence automatically update the map.

This paradigm requires all the three fundamental components described above, a digital map, a way of establishing map/image correspondence, and a way of using the map to guide image analysis. It therefore provides a good testbed for both the underlying concepts and their implementation.

* The map must be represented in a form that allows inherent geometrical and topological properties to be retrieved readily. It must also permit storage of symbolic information that goes beyond what is contained in conventional maps.

* Determining and using the correspondence among photo, map, and scene is a crucial part of our approach. Photogrammetry and AI research have together given us most of the necessary computational tools. The one remaining gap is determining which map and photo features correspond. For the time being, we rely on the user to bridge that gap, and we employ established algorithms to do the rest.

* Guided tracing of a linear feature, such as a road or coastline, is a particularly clear way in which knowledge in the form of a map may be used to guide analysis of an image.

3. Data

In order to provide a coherent problem area for our research, we have selected a single geographical area and obtained comprehensive photo coverage of it for use as our primary set of data. The area we have selected is centered upon Oakland, California. Included in this area are Alameda Naval Air Station, Treasure Island Naval Reservation, Oakland Army Terminal, Oakland Naval Supply Center, the cities of Oakland and Berkeley, and the mountains behind them. Also in this area are railyards, harbors, airfields, bridges (including the Bay Bridge), freeways, urban areas, open hillsides, lakes, and streams.

We have obtained photo coverage from the US Geological Survey, including:

* NASA Skylab coverage of the entire San Francisco Bay area, taken from an altitude of 250 miles.
* U2 coverage of the Oakland area - stereo pairs of vertical and oblique views, taken from 60,000 feet.
* High altitude mapping photography, taken from 40,000 feet.
* Medium altitude mapping photography, taken from 15,000 feet, giving comprehensive overlapping coverage of the Oakland area.

In addition, we have obtained detailed USGS maps of the entire Oakland and San Francisco Bay areas. We have also obtained Digital Terrain Data tapes which give an array of ground elevations for this area; this terrain data was originally compiled by the Defense Mapping Agency.

From 544th Squadron, Strategic Air Command, we have received a set of (unclassified) reconnaissance photographs of various locations within the United States. These photographs provide representative examples of several common photointerpretive problems, including change detection, box-car classification and counting, airfield measuring, and storage container capacity measurement.

4. System Building

We are constructing a basic system to provide a framework for experimentation.

a. Requirements

The system must provide the following capabilities:

* Basic facilities for handling large digitized images.
* Basic image processing routines.
* Storage of maps and symbolic information in an appropriate data structure.
* Interactive facilities for adding to or modifying the map.
* Coordinate transformation routines for establishing and using correspondences among picture, map, and scene.
* Techniques for using map information to guide image processing.

b. System Structure

The structure of the basic system is shown in block diagram form in Figure 3. It is currently implemented as two communicating forks under the TENEX operating system. One fork is written in SAIL, for efficient numerical and array operations, and contains image handling, processing, and displaying routines. The other fork is written in INTERLISP, for symbol manipulation, and contains the map data structure, the routines for manipulating it, the interactive interface, and the main control routines.

c. Map Representation and Use

Maps are represented by a network data structure, which we call an Association Net. It is similar in organization to a Semantic Net, but differs in two ways: there is only one type of object -- an

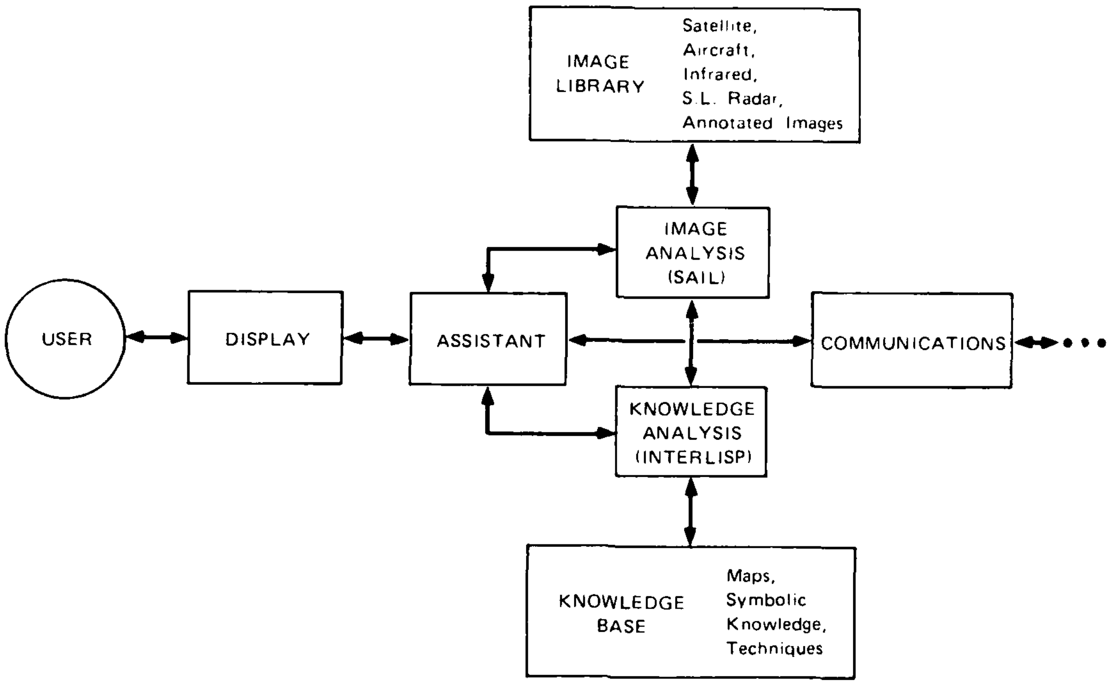


FIGURE 3 BLOCK DIAGRAM OF EXPERIMENTAL SYSTEM

association -- instead of two -- objects and relations: it is composed of direct pointers, instead of names that stand for objects. Associations are represented by lists, each of which has an indicator and a value. For example, (TYPE . ROAD) has TYPE as indicator and ROAD as value. The value may be a list, and in particular, a list of associations. For example,

(ARC (NAME . MAINSTREET) (TYPE . ROAD) (WIDTH . 50) ...)

Networks of roads are represented as networks in the data base. Significant points, such as intersections or bends, are represented by data objects with indicator NODE. Significant linear segments, such as portions of coastline or roads, are represented by objects with indicator ARC. Each NODE has an association POINT which gives it real world location, and as many ARCs as line segments

radiating from it (frequently two for bends). For convenience, a linear segment is actually represented by two ARCs, each associated with one end point and with the other arc, as in Figure 4. This interlinked structure makes it particularly easy to trace a road by stepping through a sequence of ARCs and NODEs. It also facilitates editing by allowing the insertion of new nodes without radically disturbing the existing data structure.

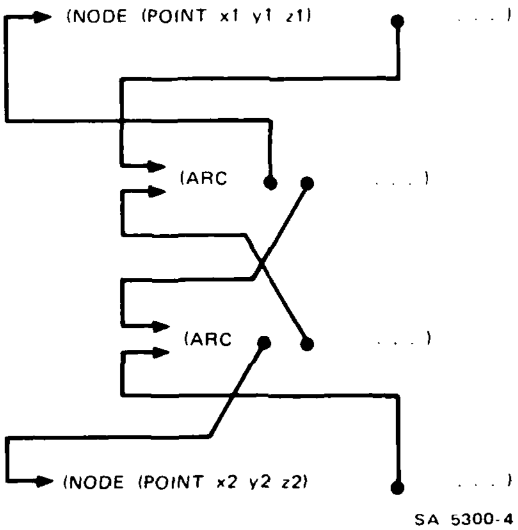


FIGURE 4 REPRESENTATION OF A LINE SEGMENT

Each node and arc has an associated name and type. The name allows rapid indexing into the map structure, and following of named features, such as following roads through intersections. A type is an association list, like a node or an arc, which has an associated mode (line or point), a function for displaying objects of this type, and associated display color. The types currently in the system are: City, Bridge, River, Lake, Coast, Tank, Intersection, Building, Boundary, Cpoint, Tpoint and Invis. The latter three types are not displayed but are used for crossings of linear features, through points and conceptual linking of entities, respectively.

The nodes and arcs are contained in a "World" association list that represents the entire map. The world contains association lists of nodes and arcs according to type, thus giving a means of access to the entire association net for purposes such as creating a thematic overlay displaying all rivers.

A simple example of a fragment of a map is shown in Figure 5.

d. Image Representation and Use

The photographs we have collected are of very high quality, having an effective resolution of about 20000x20000 pixels. This poses something of a data conversion and storage problem: If we were to completely digitize every picture at the resolution necessary for detailed analysis of small features, the process would take months and would require hundreds of reels of magnetic tape. Our solution was to digitize * the pictures completely at the manageable resolution of 1000x1000 pixels of 8 bits of (logarithmic) density, and then to digitize selected picture fragments at higher resolution as required. This simulates a future operational system in which the original photographs are used as primary storage and fragments are digitized on-line, on demand.

A single picture at 1000x1000 resolution fills 256k of PDP-10 core, so that we are processing one 256x256 subimage at a time. We have written the necessary software to sample, reformat, file, and display pictures in this manner. We can, for example, sample the entire picture and display it on the 256x256 Ramtek display. We can then indicate an area interactively with the cursor and have the picture resampled and redisplayed so that the selected area now fills the entire screen.

----

* High quality image digitization was performed on an Optronics scanner at the Image Processing Institute, University of Southern California. We are grateful to Professor W.K. Pratt and his staff for their cooperation.

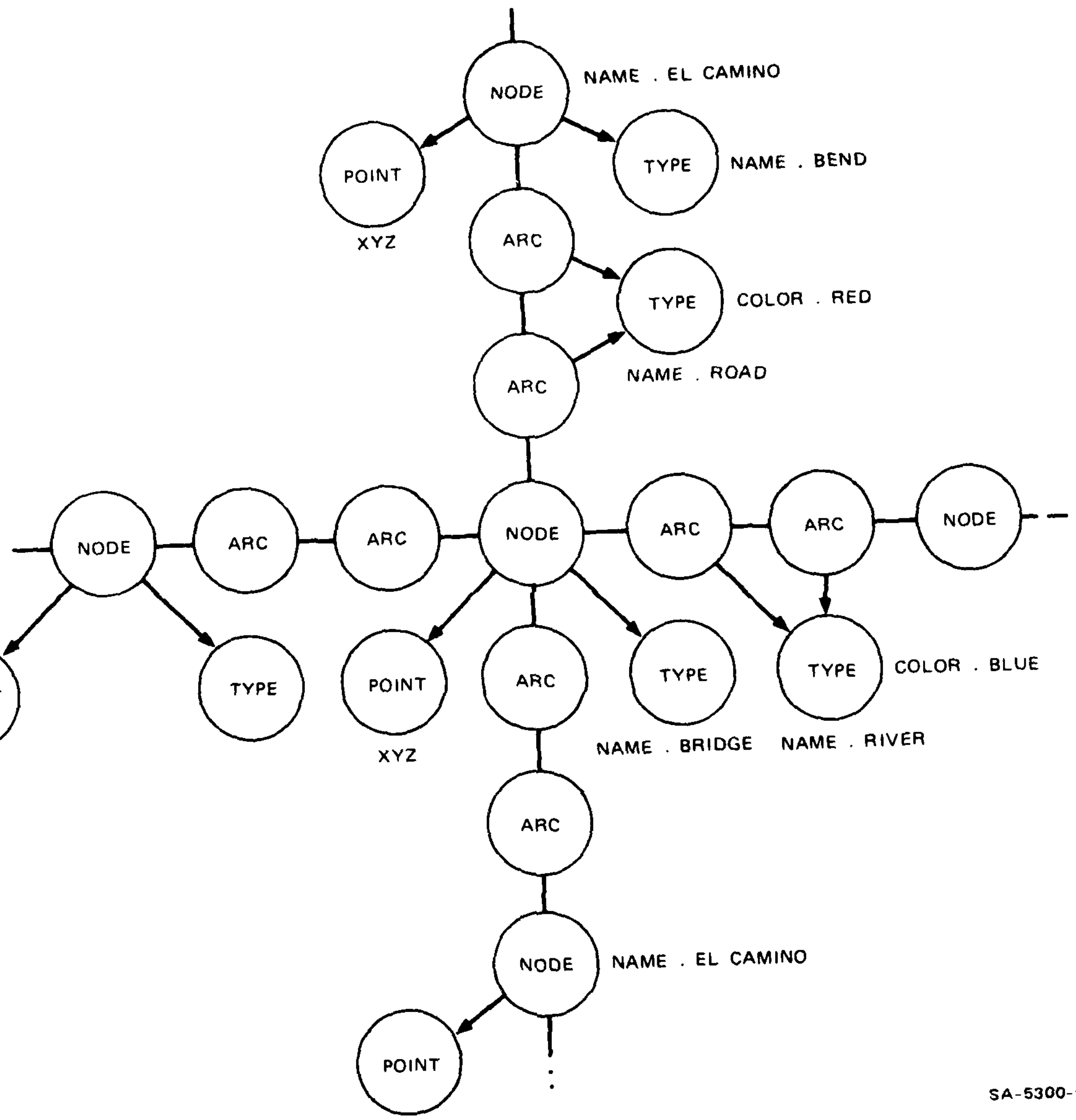


FIGURE 5 A SIMPLIFIED MAP FRAGMENT

Interactive routines have been written for using the display trackball to indicate points or draw lines overlaid on the displayed gray-scale picture. They incorporate the necessary coordinate transformation routines so that locations on the display screen may be measured and referred to in terms of full picture coordinates (rather than subimage coordinates).

We have also provided ourselves with basic picture processing primitives which operate on 256x256 arrays (of arbitrary byte size). Available functions include logical primitives that operate on 1-bit arrays, and operations for level slicing, scaling, and determining statistics, such as minimum and maximum values or histogram. More sophisticated functions include a routine for applying an arbitrary operator to points within a specified window and indicated by a 1-bit mask array, an edge-tracker that traces the boundary of a region designated by a predicate, and a line- fitting routine that approximates a traced boundary by a sequence of straight line segments.

All of the picture processing functions are encoded in SAIL and operate in a lower TENEX fork. Appropriate routines have also been written in LISP which interface to the SAIL routines so that they may be called by high level application-specific functions, or interactively from the user's console.

5. Initial Experiments

We are working toward the demonstration paradigm described earlier by constructing a crude, but complete, system and then refining expertise of the various components.

a. Completed Facilities

The current state of the implementation is best indicated by an example.

The user can retrieve an aerial photograph, and have it sampled and displayed at coarse resolution (Figure 6). He can then use

the cursor to indicate points and draw lines superimposed on the picture. For example, he can trace a road with the cursor, or indicate locations of intersections or buildings. Linear features are traced as a sequence of line segments and can be backed up and redrawn until the user is satisfied as to accuracy and appearance (Figure 7).

After the user has designated a name and type, the traced feature is automatically entered into the map data base. For example, in drawing a minor road linking two previously traced major roads, the end points of the new road will be linked to nodes lying on the existing roads, if suitable ones exist. If no existing nodes are sufficiently close to the end points, new nodes will be automatically inserted into the existing roads. The updated map is then redisplayed over the image.

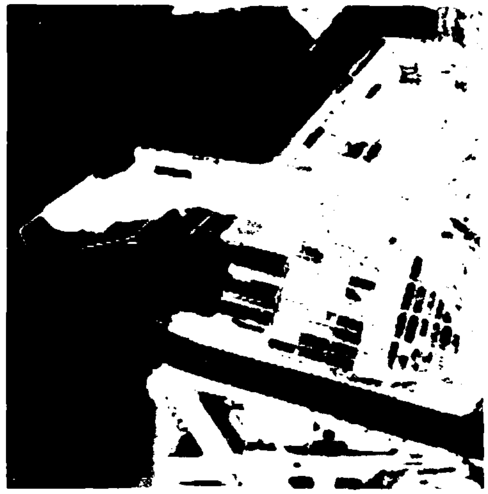
The user may now select an area for closer scrutiny by indicating it with the cursor on the display (Figure 8). The picture file is sampled appropriately and the selected area is displayed to fill the screen. A clipping routine determines which line segments in the map cross the displayed area, and displays those that are visible. At the same time, an index to the displayed nodes and arcs is created so that it is possible to identify rapidly which is being pointed at with the cursor (Figure 9).

The detailed display may now reveal that the earlier tracing on the coarse display was inaccurate. The user may then point at a node and have it moved to a new location, to which he also points. Nodes may also be inserted into existing arcs, to better approximate the shape of a curve, for example, and nodes or arcs may be deleted. Figure 10 shows the result of deleting an erroneous fragment of the map, while Figure 11 shows the insertion of a better fragment. Finally, Figure 12 shows the result of a sequence of editing operations in the selected area, and Figure 13 shows the entire map.

The editing routines maintain the integrity of the data base in these operations. It is also possible for the user to undo his actions, by using the LISPX command "UNDO". In this way he not only can

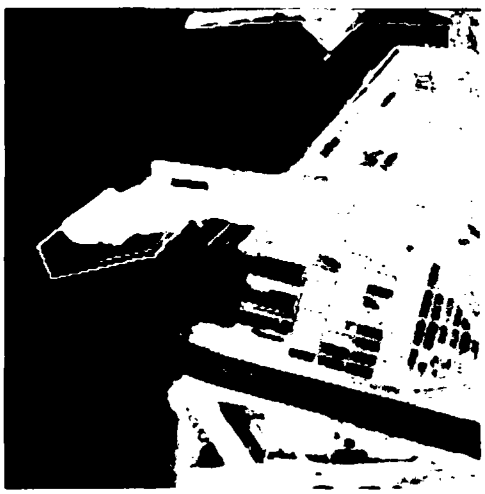
SA-5300-10

FIGURE 6 FULL PICTURE AT 256 x 256 RESOLUTION



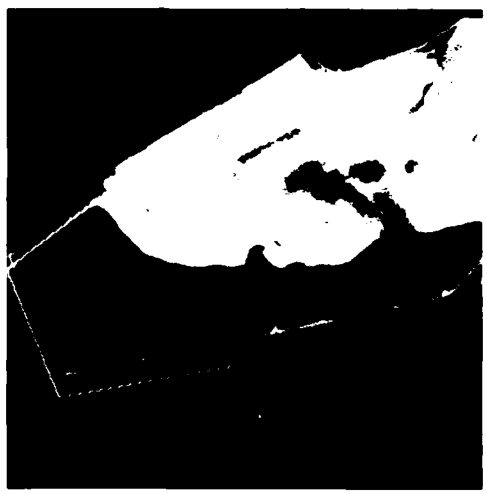
SA 5300 9

FIGURE 7 COARSE MAP TRACED ON PICTURE



SA 5300 8

FIGURE 8 AREA SELECTED FOR DETAILED EXAMINATION



SA 5300 7

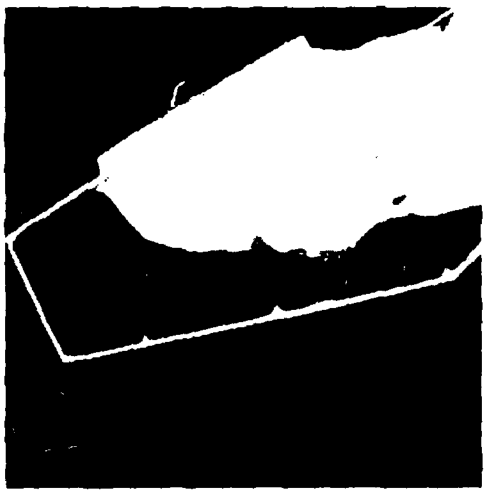
FIGURE 9 AREA DISPLAYED AT FULL RESOLUTION

SA 5300-15

FIGURE 10 DELETION OF AN ERRONEOUS PIECE OF MAP

SA 5300-16

FIGURE 11 INSERTION OF A BETTER TRACE



SA 5300-17

FIGURE 12 RESULT OF EDITING MAP

SA 5300-18

FIGURE 13 THE MAP AT COARSE RESOLUTION

undo mistakes but also can freely construct temporary data structures for use in specialized processing later without worrying about difficulties in cleaning them up.

The different areas displayed by the user are remembered by the system, together with names that he gives them. He may thus return to a previously displayed area, such as the initial overall view, for reexamination or modification of the map.

In situations in which the feature being traced has good contrast, the user can call upon a simple automatic tracing routine. This routine tracks round the boundary of a region defined by any arbitrary predicate. The current default predicate is comparison with a threshold brightness, such as "Darker than threshold". The tracing routine must be provided by the user with a starting location within the region (usually via the cursor) and an initial direction (Figure 14). It steps in the specified direction until the predicate is no longer satisfied, and then tracks along the boundary between satisfaction and non-satisfaction.

The result of the edge tracer is a chain-encoded boundary of the region (Figure 15). This boundary is then passed to a line-fitting routine which approximates the boundary by straight line segments. For cartography we wish to represent the boundary by line segments that are nowhere more than a specified distance from the real boundary: in a conventional map, no feature may be more than 50 feet in error. This approximation is achieved by building up the line segments incrementally. Fixing one end point of the segment, the other end is stepped along the boundary one point at a time. For each position, the straight line joining the end points is computed, and the deviations of the intervening points from the line are also computed. If the maximum of these deviations is less than the permitted error, we step the end one point further and repeat. If the maximum deviation is greater than the permitted error, we have stepped too far. In this case, the previous point, or alternatively the point that possesses the maximum

FIGURE 14 START OF BOUNDARY TRACING

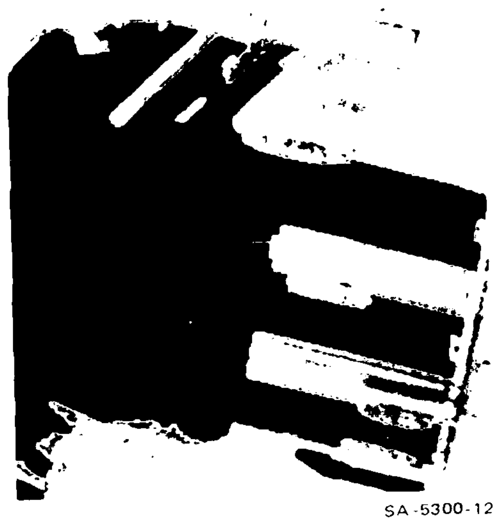


FIGURE 15 AUTOMATIC TRACE OF BOUNDARY

FIGURE 16 AUTOMATIC LINE FITTING AND INSERTION INTO MAP

FIGURE 17 RESULT OF EDITING THE TRACED BOUNDARY

deviation, is taken as the end point of this segment, and also as the start of the next segment. The stepping and testing are repeated until the entire boundary has been fitted to the desired accuracy; the result of the fitting routine is a list of the line segments found (Figure 16).

The line segments found by the tracing and fitting routines are then automatically incorporated into the map data structure. Because of the elementary nature of the tracing routine, the boundary may be in error in several places. The user can, however, edit the map and redisplay the improved boundary (Figure 17). Despite the very early stage of this work, the semi-automatic tracing and editing process is much quicker than manually tracing the boundary with the cursor.

b. Work in Progress

The current work is concerned with two problems: coordinate transformations and smarter tracing.

At present, the coordinates of points in the map are simply the coordinates of the feature in the full 1000x1000 picture. The only transformation we currently handle is the simple one between full picture and displayed subimage. We are thus simply tracing on the picture without considering the true relation between the picture and the ground, and without possessing the ability to use multiple pictures. We are now in the process of testing the necessary routines to correct these defects.

When the coordinate transformation routines are integrated into the system, the user will be able to point at several landmarks in the picture, giving their world locations if they are not already in the map. The system will then compute the coordinate transformation between picture and world by performing a numerical optimization of the parameters of a camera model, which includes location, heading, altitude, pitch, and roll. Once this optimization has been done, it will be possible to represent world coordinates in the

map, to employ multiple pictures and correctly display a windowed portion of the map overlaid on the picture, to point at features in the picture and obtain their world locations, and to measure distances between two indicated points.

The current work on automatic tracing is directed at two problems: devising techniques for detecting and locating different kinds of linear features, and using approximate tracings to guide the detector.

The linear features with which we are concerned may be one-sided, such as a lake boundary, or two-sided, such as a road, and the two cases require slightly different treatment. For example, in road tracing, if one edge becomes hard to detect, the other may still reveal the course of the road. Generally speaking, knowledge about the type of feature being traced and statistics of what has so far been found may be used to tailor the detector and greatly improve its performance. We are currently studying the general characteristics of roads -- which are far from being the ideal of a uniform line on a uniform background. Existing edge detectors do not appear to be optimally matched to real roads.

We are considering several ways of guiding the tracing. One possibliity is to apply the detector generally, over the region where the road may be, and to weight its output according to distance from the approximate tracing. The problem then becomes one of finding the path for which the sum of the weighted outputs is greatest. Relatively little work has been done previously on guided tracing, especially in the complex and variable domain of aerial photographs.

When guided tracing is an integral part of our system, it should be possible for the user to sketch features roughly and for the system to trace them accurately with reasonable reliability. Features that are already in the map may be found and traced entirely automatically. A natural extension of this technique will allow determination of the coordinate transfomation more automatically.

D. Conclusions

1. Applications Requirements

The most important findings of the work reported here are that many problems in Cartography and Photointerpretation are bottlenecks in production and require the approaches of Machine Vision to automate them.

In Cartography, the primary need is for automatic tracing of features in aerial photographs, both of new features and of features on existing maps. This need in turn requires use of knowledge of the characteristics of features traced and accurate registration of map and picture for photogrammetric purposes.

In Photointerpretation, specialized aids, such as for counting, measuring, and change monitoring, are required to alleviate the photointerpreter's burden. Such capabilities must draw upon a great amount of knowledge from the map data base to perform reliably.

The common requirements of the two domains thus include establishment of the correspondences among map, picture, and scene, and the exploitation of diverse types of knowledge to guide image analysis.

The current techniques of Machine Vision appear adequate for dealing with some of these tasks at a basic level. We recognize that complete automation is some years away; the optimal approach is via an interactive system that can employ advice offered by the user to guide its activities.

2. Practical Issues

We have already encountered some practical impediments which are not within the immediate scope of the current and proposed research, but which must eventually be removed to accomplish applications objectives. In the meantime, we circumvent them to minimize their drain on our resources.

The quantity of information inherent in a high resolution aerial photograph is too great to handle entirely digitally with ease. The solution is to use photographs as primary storage and digitize pieces of them on demand. We are simulating this solution.

The low level operations on pictures are simple but time consuming. Many of them could be readily accomplished by special-purpose hardware, yielding a speed increase of several orders of magnitude. In the absence of such hardware, we perform all operations sequentially on the computer.

The programming languages currently available do not support application-oriented image understanding: both compilation of efficient code and handling of symbolic information are necessary, and no single language provides both. In addition, very large addressing spaces are needed. For the time being, we employ several communicating forks, written in different languages. This compromise enables us to exploit the best features of several different languages to some extent, but costs considerable overhead in programming, redundant representation, and data communication.

3. Scientific Issues

The central scientific issue in the above applications, and in Machine Vision in general, involves the role of knowledge. It is evident that the more knowledge that is brought to bear on a problem, such as road tracing, the better the performance can be. The nature of the knowledge employed and the manner of its employment are open questions which must be answered before high performance image understanding systems can be constructed. Resolution of these issues is crucial to the attainment of more fully automatic cartography and photointerpretation.

REFERENCES

1. H. G. Barrow, "Interactive Aids for Cartography and Photointerpretation", Proposal SRI No. ISU 76-46 submitted to Defense Advanced Research Agency, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (February 1976).

2. "Specifications for Military Maps", TMS-1, Defense Mapping Agency Topographic Center (September 1974).

3. "Digital Image Processing", Digital Image Systems Division, Control Data Corporation, Minneapolis (September 1974).

4. T. D. Garvey and J. M. Tenenbaum "Stratefies for Purposive Vision", Tech. Note No. 117, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (1975).

5. J. M. Tenenbaum et al., "ISIS: An Interactive Facility for Scene Analysis Research", Tech. Note No. 95, Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (1974).

6. T. D. Garvey and J. M. Tenenbaum, "Application of Interative Scene Analysis Techniques to Cartography", Artificial Intelligence Center, Stanford Research Institute, Menlo Park, California (Tech. Note to be published).

Appendix I

CARTOGRAPHIC AND PHOTO INTERPRETIVE REFERENCES

Appendix I

CARTOGRAPHIC AND PHOTO INTERPRETIVE REFERENCES

I. General Reference Material

A. Reference Books

1. Manual of Remote Sensing, Leonard W. Bowden (ed.), published by American Society of Photogrammetry, Falls Church, Virginia, 1975.

2. Manual of Photogrammetry, Robert N. Colwell (ed.), published by American Society of Photogrammetry, Falls Church, Virginia, 1960.

B. Major Journals

1. Photogrammetric Engineering and Remote Sensing

C. Major Conference Proceedings

1. American Society of Photogrammetry Falls Church, Virginia (Annual)

2. American Congress on Surveying and Mapping (Annual), Woodward Bldg., Room 430, 733 15th Street N.W., Washington, D.C. 20005

II. Department of the Army Technical Manuals

A. GEODETIC AND TOPOGRAPHIC SURVEYING, TM 5-441, February 1970.

B. ENGINEER INTELLIGENCE, FM 5-34.

C. CARTOGRAPHIC AERIAL PHOTOGRAPHY, TM 5-243, January 1970.

D. ENGINEERS' REFERENCE & LOGISTIC DATA, FM 5-35.

E. FOREIGN MAPS, TM 5-248, October 1963.

F. FIELD ARTILLERY CANNON GUNNERY, FM 6-40.

G. MILITARY GEOGRAPHIC INTELLIGENCE (TERRAIN), FM 30-10, March 1972.

H. MILITARY SYMBOLS, FM 21-30.

I. COMBAT INTELLIGENCE, FM 30-5, October 1973.

J. STAFF OFFICER FIELD MANUAL, FM 101-10.

K. ROUTE RECONNAISSANCE AND CLASSIFICATION, FM 5-36, January 1970.

L. ELEMENTS OF SURVEYING, TM 5-232.

M. TOPOGRAPHIC SYMBOLS, FM 21-31, June 1961.

N. MAPPING FUNCTIONS OF THE CORPS OF ENGINEERS, TM 5-231.

O. COMPILATION AND COLOR SEPARATION OF TOPOGRAPHIC MAPS, TM 5-240, June 1971.

P. TERRAIN MODELS & RELIEF MAP MAKING, TM 5-249.

Q. SPECIFICATIONS FOR MILITARY MAPS, TM 5-1, September 1974, revised December 31, 1975.

R. TACTICAL INTERPRETATION NOTEBOOK, TM 30-246.

S. PHOTOGRAPHIC, TM 30-245.

T. MAP READING, FM 21-26.

III. DMA Reports

A. COMMONWEALTH SURVEY OFFICERS CONFERENCE 1975, DMA AUTOMATED CARTOGRAPHIC SYSTEMS by Robert A. Penney.

B. OFF-LINE ORTHOPHOTO PRINTING AT THE DEFENSE MAPPING AGENCY TOPOGRAPHIC CENTER by Hayne B. Dominick.

C. COLOR SEPARATION SYMBOLIZATION IN SEMIAUTOMATED MAP PRODUCTION by William H. Burdette.

D. DIGITAL TOPOGRAPHIC INFORMATION BANK by Henry R. Cook.

E. MAKING A MAP WITH DIGITAL DATA by Robert L. Struck.

F. AUTOMATED DELINEATION OF GROUND SLOPE by Merle J. Biggin.

G. PROBLEMS, SHORTFALLS, AND NEEDS OF TOPOGRAPHIC MAPPING by Reuben Cook.

IV. US Army Engineer Topographic Laboratories' Reports

A. DISPLAY TECHNOLOGIES FOR TOPOGRAPHIC APPLICATIONS. ASSESSMENT OF STATE-OF-THE-ART AND FORECAST, June 1975.

B. PARALLEL OPTICAL PROCESSING TO CONVERT ELEVATION DATA TO SLOPE MAPS, PHASE II: PRACTICAL CONSIDERATIONS, February 1975.

C. COMPUTING A LINE-OF-SIGHT USING DIGITAL IMAGE MATCHING AND ANALYTICAL PHOTOGRAMMETRY, March 1975.

D. SURFACE MATERIALS AND TERRAIN FEATURES OF YUMA PROVING GROUND, May 1975.

E. A TERRAIN EFFECTS ANALYSIS ROUTINE FOR AN MGI SYSTEM, April 1975.

F. TEXTURE TONE STUDY: SUMMARY AND EVALUATION, March 1975.

G. A SYSTEM FOR TOPOGRAPHIC INQUIRY NO. 3 ALPHANUMERIC SUBSYSTEM DATA BASE LISTING, March 1975.

H. A MATRIX EVALUATION OF REMOTE SENSOR CAPABILITIES FOR MILITARY GEOGRAPHIC INFORMATION (MGI), July 1972.

I. PARALLEL OPTICAL PROCESSING TO CONVERT ELEVATION DATA TO SLOPE MAPS, PHASE I, THEORETICAL ANALYSIS, October 1974.

J. TRANSFORMATION OF COORDINATES OF CARTOGRAPHIC DIGITAL DATA, October 1974.

K. HISTORY OF U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES (1920 to 1970), November 1973.

L. ANALOG GRAPHIC PROCESSING FOR 3-D TERRAIN DISPLAYS, PROFILES, AND ELEVATION LAYER TINTS, October 1975.

M. PRELIMINARY IMAGE DATA EXTRACTION EXPERIMENTS WITH THE PHASE I, AUTOMATED IMAGE DATA EXTRACTION SYSTEM-I, December 1974.

N. TEXTURE TONE STUDY: SUMMARY AND EVALUATION, March 1975.

V. Other

A. COMPASS PREVIEW--A NEW SYSTEM FOR EXPLOITATION OF IMAGERY FOR INTELLIGENCE, Air Force Systems Command, Rome Air Development Center, Intelligence Reconnaissance Division, Intelligence Data Handling Branch, Griffiss AFB, NY 13441.

B. AIR SPYING, Constance B. Smith, 1947.

Appendix II

CONTACTS WITH CARTOGRAPHIC AND PHOTOINTERPRETIVE CENTERS

Appendix II

CONTACTS WITH CARTOGRAPHIC AND PHOTOINTERPRETIVE CENTERS

I Defense Mapping Agency Topographic Center, Washington, D.C.

D. Meier, Office of the Director

II U.S. Army Engineering Topographic Labs., Fort Belvoir, Virginia

H. Carr, Chief, Autocartography Group
L. Gambino, Chief, Computer Science Lab
B. Scheps, Chief, Technology Development Branch

III U.S. Geological Survey, Reston, Virginia

M. McKenzie, Chief of Photogrammetry

IV U.S. Geological Survey, Menlo Park, California

A. Stein, Photogrammetry
D. Edson, Autocartography

V Strategic Air Command, Offutt Air Force Base, Nebraska

Col. J.L. Passauer, Vice Commander, 544th Aerospace Reconnaissance Technical Wing
Maj. T. Profett, Air Force Global Weather Center

VI Rome Air Development Corp., Griffis Air Force Base, New York

R. Hoffman
Maj. J. Broglie

VII Central Intelligence Agency, Washington, D.C.

L.F. Wise

VIII Aeronutronics Ford, Palo Alto, California

R. Asendorf
S. Fraelick
R. Widergren

IX. Lockheed Missiles and Space Co., Palo Alto Research Labs, Palo Alto, California

M.A. Fischler

Appendix III

FILE STRUCTURES FOR COMMAND AND CONTROL DATA BASE

Appendix III

FILE STRUCTURES FOR COMMAND AND CONTROL DATA BASE

Following is a complete description of the six files that comprise the command and control data base in Datalanguage format. It includes in the comments an explanation of the meaning of the fields, and in some cases, some typical examples of their values.

PRECEDING PAGE BLANK-NOT FILMED

```
CREATE SHIPFILE FILE LIST (1,200,1000)
 SHIPRECORD STRUCTURE
  NAME STRING (26),I=D          /* SHIP NAME                        */
  CLASS STRING(15),I=D          /* POINTER TO CLASS RECORD          */
  FLAG STRING (2),I=D           /* COUNTRY                          */
                        /*VALUES: US,UR(USSR),UK,NE(THERLANDS),AR(GENTINA) */
                        /*SF(SOUTH AFRICA),WG(WEST GERMANY),SA(UDI ARABIA) */
                        /*LI(BERIA),AN(GOLA),NO(RWAY),VE(NEZUELA),FR(ANCE) */
                        /*IT(ALY),SP(AIN),PO(RTUGAL),CA(NADA),EG(YPT)      */
  CAT STRING(3),I=D     /*CATEGORY; VALUES ARE:MER(CHANT),FSH(FISHING)     */
                        /*      NAV(AL)                                    */
                 /*  TYPE       SHIP TYPE IN ACRONYM FORM; VALUES ARE:     */
                        /*AGI(INTELLIGENCE COLLECTOR),AO(FLEET OILER)      */
                        /*AOE(OILER AND AMMUNITION SHIP),                  */
                        /*BULK(CARGO FREIGHTER),CG(GUIDED MISSILE CRUISER  */
                        /*CLG(GUIDED MISSILE LIGHT CRUISER)                */
                        /*CLGN(NUCLEAR POWERED GUIDED MISSILE LIGHT CRUISER*/
                        /*CV(AIRCRAFT CARRIER),DD(DESTROYER)               */
                        /*DDG(GUIDED MISSILE DESTROYER),FF(FRIGATE)        */
                        /*FFG(GUIDED MISSILE FRIGATE),SS(ATTACK SUBMARINE) */
                        /*SSN(NUCLEAR ATTACK SUBMARINE)                    */
                        /*SSBN(NUCLEAR POWERED BALISTIC MISSILE SUBMARINE) */
                        /*SSGN(NUCLEAR POWERED GUIDED MISSILE SUBMARINE)   */
   TYPE1 STRING(1),I=D  /*VALUES ARE:A(INTELLIGENCE COL. OR OILER),B(ULK)  */
                        /*C(RUISER OR CARRIER),D(ESTROYER),F(RIGATE),S(UB) */
   TYPE2 STRING(1),I=D  /*VALUES ARE:D(ESTROYER),F(RIGATE),L(IGHT CRUISER) */
                        /*G(INT. COL. OR GUIDED MISSILE),O(ILER),S(UB),    */
                        /*U(BULK),V(AIRCRAFT CARRIER)                      */
   TYPE3 STRING(1),I=D  /* VALUES ARE:B(SSBN),E(AOE),G(UIDED MISSILE)      */
                        /*L(BULK),N(UCLEAR)                                */
   TYPE4 STRING(1),I=D  /* VALUES ARE:K(BULK),N(NUCLEAR)                   */
   TYPE STRING(4),VE=TYPE1!TYPE2!TYPE3!TYPE4,I=D
   HUL STRING (4),I=D   /*HULL NUMBER                        */
   IRCS STRING(6),I=D   /*INTERNATIONAL RADIO CALL SIGN      */

          /* *** END OF SHIP IDENTITY PARAMETERS  ***                      */

   DOCTR STRING(1)      /*VALUES ARE: D IF DOC IS KNOWN TO BE ON BOARD.    */
                        /* * O/WISE; USED IN SEARCH/RESCUE TYPE OPERATIONS */
   TITLE STRING(18),I=D /*TITLE OF CONVOY; EXAMPLES:QC50X, NEW YORK-LONDON */
   PCFUEL INTEGER(3),S=ASCII,R=DEC /* PERCENTAGE OF MAXIMUM FUEL THE SHIP  */
                        /*HAS ON BOARD. WHEN NOT REPORTED, THIS FIELD IS   */
                        /*CALCULATED; IT IS 100 FOR NUCLEAR POWERED OR     */
                        /* MERCHANT SHIPS                                  */

                 /* *** END OF COMMON PART *** */
```

```
USNAVALFLG INTEGER(1),S=ASCII,R=DEC /*1 IF IT IS A US NAVAL SHIP;0 O/WISE*/
USNAVALPART LIST(0,1),C=USNAVALFLG /*NEXT PART IS FOR US NAVAL SHIPS ONLY*/
  USNAVRECORD STRUCTURE
    HOGEO STRING(4)     /*GEOGRAPHIC LOCATION CODE OF PLACE AT WHICH AN     */
                        /*ORGANIZATION IS PERMANENTLY ATTACHED; EXAMPLES:   */
                        /*NORF(OLK),MAY(PORT),CHAR(LESTON),JACK(SONVILLE)   */
                        /*OCEA(NA)                                          */
    CONAM STRING(22)                         /*NAME OF COMMANDING OFFICER  */
    LINEAL INTEGER (9),S=ASCII,R=DEC,F=' ' /* LINEAL   OFCOMMANDING        */
                                 /*OFFICER;LOWER   IMPLIES HIGHER SENIORITY */
    OPCON STRING (30) /* ABBREV. NAME OF CONTROLLING ORGANIZATION           */
                        /*IF NON-BLANK, AND NO POSITION FIELDS PRESENT,THEN*/
                        /*SHIP IS WITH SHIP IN WHICH THE OPCON ORG. IS;     */
                        /*IF NO POSITION IS SHOWN FOR THAT SHIP, THEN IT IS*/
                        /*WITH THE SHIP CARRYING THE NEXT SENIOR COMMANDING*/
                        /*OFFICER;EXAMPLE:THE SOUTH CAROLINA WITH CTU27.7.1*/
                        /*EMBARKED WOULD THEN BE WITH THE JOHN F. KENNEDY   */
                        /*WHICH HAS CTG27.7 EMBARKED                        */
    READY INTEGER (1),S=ASCII,R=DEC /*CURRENT STATE OF READINESS; VALUES */
                        /*RANGE FROM 1 TO 5: 1 IS TOTALLY READY .          */
                        /* SHOULD BE DISPLAYED AS C1,C2,C3,C4 OR C5        */
    REASN STRING(1)     /* REASON WHY READY IS NOT C1; POSSIBLE VALUES ARE:*/
                        /*E(AMMUNI. SHORT.),C(FOOD SHORT.),F(UEL SHORTAGE) */
                        /*G(UN SYST. FAILURE),H(ULL DAMAGE),M(ISSILE SYST. */
                        /* FAILURE),P(ROPULSION CASUALTY),R(SURF.SEARCH    */
                        /* RADAR),S(ONAR FAILURE),T(ORPEDO SYST.FAILURE)   */
                        /*A(AIR SEARCH RADAR FAILURE),O(VERHAUL)           */
    EIC STRING(7)       /*CODE ASSIGNED TO A SPECIFIED PIECE OF EQUIPMENT; */
                        /*EXAMPLES:AJ643                                   */
    CASREP INTEGER(10),S=ASCII,R=DEC /*TIME OF THE CASUALTY MESSAGE        */
    CARAT INTEGER (1),S=ASCII,R=DEC  /*PROJECTED STATE OF READYNESS        */
    CADAT INTEGER (6),S=ASCII,R=DEC  /*PROJECTED DATE FOR CARAT            */
    ETERM STRING (10)       /*EMPLOYMENT TERM; VALUES ARE:         */
                        /*ANTSHIP(ANTI SHIPPING OP.),ASOPS(ANTI SUB OP.)   */
                        /*CARESC(CARRIER ESCORT),CONVESC(CONVOY ESCORT)    */
                        /*OVHL(OVERHAUL),RAV(RESTRICTED AVAILABILITY)      */
                        /*REPL(REPLENISHMENT UNDERWAY),STROPS(STRIKE OP.)  */
                        /*SURVOPS(SURVEILLANCE OP.),TAV(TENDER AVAILABILI.)*/
                        /*TRANS(IT),TRNG(AT SEA TRAINING),UPKEEP(ROUTINE   */
                        /*MAINTENANCE IN PORT)                             */
    EBEG INTEGER(6),S=ASCII,R=DEC /*EMPLOYMENT STARTING DATE,PAST OR FUT.*/
    EEND INTEGER(6),S=ASCII,R=DEC /*EMPLOYMENT ENDING DATE                 */

    END   /* *** END OF PART VALID ONLY FOR US NAVAL SHIPS ***    */
```

```
MERFLG INTEGER(1),S=ASCII,R=DEC /*1 FOR MERCHANT SHIPS, 0 O/WISE          */
MERONLYPART LIST(0,1),C=MERFLG /*NEXT PART PRESENT ONLY IN MERCHANT SHIPS*/
  MERONLYRECORD STRUCTURE
    HIT STRING(1),I=I /* H INDICATES SHIP HAS BEEN DESIGNATED AS A HIGH  */
                      /* INTEREST TARGET; IT IS * O/WISE                 */
    PASS INTEGER(4),S=ASCII,R=DEC     /*  OF PASSENGERS ON BOARD         */
    CARGOTYPE STRING(6)        /* TYPE OF CARGO; VALUES ARE: OIL,UNK(NOWN)*/
                      /*COAL,TIN,TANKS,TUNGST(EN),PHOS(PHATES)           */
                      /*VNAD(VANADIUM ORE),CHRORE(CHROME ORE),TRUCK(S)   */
                      /*FARMAC(FARM MACHINERY),ACFT(AIRCRAFT),FOOD,WHEAT */
                      /*CONST(RUCTION),GENMER(GENERAL MERCHANDISE)       */
                      /*MATOOL(MACHINE TOOLS),AMMO(AMMUNITION)           */
    CARGOQTY STRING(5)         /* QUANTITY OF CARGO IN HUNDRED TONS      */
                      /*(FROM 0T TO  9999T) OR THOUSANDS BARRELS(FROM 0B */
                      /*TO 9999B). IF THE UNIT (B OR T) IS NOT INDICATED,*/
                      /*THEN, IT IS IN BARRELS FOR OIL, AND TONS O/WISE  */
   END

          /* *** END OF PART VALID FOR MERCHANT SHIPS ONLY *** */

DESTINATIONFLG INTEGER(1),S=ASCII,R=DEC /* 1 IF THE NEXT PART IS KNOWN   */
                      /*OR USEFUL; 0 O/WISE(IT IS 0 FOR FOREIGN SHIPS AND*/
                      /*ALSO FOR SHIPS IN CONVOYS)                       */
DEP STRING(18)        /*DEPARTURE NAME(FOR PORT) OR GEOCOORDINATE        */
                      /*IN THE LAST CASE: 00000-99900N; 00000-18000E     */
                      /*EXAMPLES:NORFOLK, 3700S02000E                    */
DPC STRING(2)         /* COUNTRY CODE OF DEPARTURE POINT(SAME AS FLAG)   */
ETD INTEGER(10),S=ASCII,R=DEC,F=' ' /*ESTIMATED TIME OF DEPARTURE        */
DST STRING(18)                      /*DESTINATION NAME OR GEOCOORDINATES */
DSC STRING(2)                       /*COUNTRY CODE OF DESTINATION POINT  */
ETA INTEGER(10),S=ASCII,R=DEC,F=' ' /*ESTIMATED TIME OF ARRIVAL          */

      /* *** END OF ORIGIN/DESTINATION PART *** */

TRACKCOUNT INTEGER(1),S=ASCII,R=DEC /*NEXT PART REPRESENTS THE LAST      */
                      /*POSITION OF THE SHIP IF THIS FIELD IS 1; O/WISE  */
                      /*IF THE SHIP IS IN A CONVOY: ITS POSITION IS THE  */
                      /*POSITION OF THE CONVOY; O/WISE, CHECK OPCON      */
PTP STRING(12)                /*GEOGRAPHIC COORDINATES                   */
PROB STRING (11)              /*PARAMETERS OF THE ELLIPTICAL UNCERTAINTY */
               /*AREA DEFINING THE UNCERTAINTY OF PTP; IF AREA IS CIRCULAR*/
               /*A SINGLE 3 DIGIT  IS GIVEN REPRESENTING THE RADIUS IN   */
               /*NAUTICAL MILES; IF ELLIPTICAL, 3 3-DIGIT NUMBERS ARE    */
               /*GIVEN, SEPARATED BY /'S, REPRESENTING RESPECTIVELY THE  */
               /*LENGTH OF THE SEMI-MAJOR AXIS, THE SEMI-MINOR AXIS AND  */
               /*THE ORIENTATION OF THE MAJOR AXIS RELATIVE TO TRUE NORTH */
               /*IN DEGREES; EXAMPLE:095/065/120                         */
               /*UNUSED FOR US NAVAL SHIPS AND FRIENDLY CONVOYS SINCE    */
               /*THEIR POSITION IS KNOWN                                 */
```

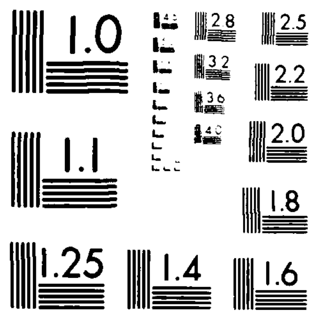
```
PTC INTEGER(3),S=ASCII,R=DEC /*COURSE,IN DEGREES CLOCKWISE FROM THE NORTH*/

PTS STRING(4)                /*SPEED IN KNOTS AND TENTHS: 00.0 TO 99.9  */
PTD INTEGER(10),S=ASCII,R=DEC /*DATE AND TIME                           */

            /* *** END OF LAST POSITION INFORMATION *** */

DRTRACKCOUNT INTEGER(1),S=ASCII,R=DEC /* IF 1 NEXT PART GIVES THE NEXT  */
                             /*ESTIMATED POSITION BY DEAD RECKONING     */
                             /*SAME REMARK AS FOR TRACKCOUNT            */
DRPOSIT STRING(12)                    /* SAME AS PTP     */
DRPROB STRING(11)                     /* SAME AS PROB    */
DRETA INTEGER(10),S=ASCII,R=DEC       /* DATE AND TIME                  */
     /* *** END OF DEAD RECKONING POSITION ***  */
END;



CREATE TRACKFILE FILE LIST (1,200,1000)
  TRACKRECORD STRUCTURE        /* ONE RECORD PER POSITION. POSITIONS MAY */
                               /* BE PAST (TRACK HISTORY), OR ESTIMATED  */
                               /* FUTURE (DRTRACK HISTORY) POSITIONS     */
  NAME STRING (26),I=D /*NAME OF SHIP, OR TITLE OF CONVOY                */
  HUL STRING (4)       /*HULL NUMBER FOR SHIP, OR 'CON' FOR CONVOY       */
  PTP STRING(12)       /*GEOGRAPHIC COORDINATES                          */
  PROB STRING (11)     /*UNCERTAINTY AREA PARAMETERS                     */
  PTC INTEGER(3),S=ASCII,R=DEC /*COURSE IN DEGREES (IS * FOR FUTURE POS.)*/
  PTS STRING(4)        /*SPEED IN KNOTS AND TENTHS (IS * FOR FUTURE POS.)*/
     /*DATE AND TIME IS SUBDIVIDED INTO YEAR-MONTH-DAY-MIN/SEC           */
  PTDYEAR INTEGER(2),S=ASCII,R=DEC,I=D        /*YEAR PART OF PTD        */
  PTDMONTH INTEGER(2),S=ASCII,R=DEC,I=D       /*MONTH PART OF PTD       */
  PTDDAY INTEGER(2),S=ASCII,R=DEC,I=D         /*DAY PART OF PTD         */
  PTDMINSEC INTEGER(4),S=ASCII,R=DEC          /*MIN-SEC PART OF PTD     */
  PTD INTEGER(10),S=ASCII,R=DEC,VE=PTDYEAR!PTDMONTH!PTDDAY!PTDMINSEC

  IND STRING(1),I=D    /*INDICATOR: IF T, THIS RECORD REFERS TO AN ACTUAL*/
                       /*PAST POSITION ( TRACK HISTORY); O/WISE IT IS D  */
                       /*AND IT REFERS TO A FUTURE ESTIMATED POSITION    */
                       /*(DRTRACK HISTORY)                               */
 END;

CREATE EMBARKEDUNITFILE FILE LIST (1,30,200),CHAPTER
 EMBUNITRECORD STRUCTURE
  ANAME STRING(1,10,30),I=D,D=' '     /*ABBREVIATED NAME FOR A UNIT    */
  CONAME STRING(1,10,22),D=' '        /*NAME OF COMMANDING OFFICER     */
  LINEAL INTEGER(1),S=BINARY,R=TWOC   /*LINEAL   OF COMM. OFFICER      */
  EMBRK STRING(1,10,26),D=' ',I=D     /*NAME OF SHIP ON WHICH UNIT IS */
```

```
  AVUNITFLG INTEGER(1),S=ASCII,R=DEC  /*1 IF AVIATION UNIT;0 O/WISE    */
      /* *** THE NEXT FIELDS ARE VALID ONLY FOR AVIATION UNITS ***     */
  AVUNITPART LIST(0,1),C=AVUNITFLG
    AVUNITRECORD STRUCTURE
      HOGEO STRING(4)          /* GEOG. LOCATION CODE OF PERMANENT BASE*/
      AIRASGN INTEGER(3),S=ASCII,R=DEC       /*  OF AIRCRAFTS ASSIGNED */
      AIREADY12HR INTEGER(2),S=ASCII,R=DEC /*  OF AIRCRAFTS AVAILABLE */
                               /*EXAMPLE: 12HR13 MEANS 13 AIRCRAFTS FOR*/
                               /* NEXT 12 HRS                          */
      AIREADY24HR INTEGER(2),S=ASCII,R=DEC  /* SAMETHING FOR 24 HRS    */
    END          /* *** END OF AVIATION UNIT PART *** */
 END;


CREATE CONVOYFILE FILE LIST (1,5,20)
  CONVOYRECORD STRUCTURE
    TITLE STRING(18),I=D                  /*TITLE OF CONVOY; EXAMPLES:QC50X  */
                                          /* OR NEW YORK-LONDON              */
    IRCS STRING(6),I=D                    /*INTERNATIONAL RADIO CALL SIGN    */
    DEP STRING(18)       /*DEPARTURE NAME(FOR PORT) OR GEOCOORDINATES        */
                         /*IN THE LAST CASE: 00000-99900N; 00000-18000E      */
    DPC STRING(2)        /* COUNTRY CODE OF DEPARTURE POINT(SAME AS FLAG)    */
    ETD INTEGER(10),S=ASCII,R=DEC,F=' ' /*ESTIMATED TIME OF DEPARTURE        */
    DST STRING(18)               /*DESTINATION NAME OR GEOCOORDINATES        */
    DSC STRING(2)                /*COUNTRY CODE OF DESTINATION POINT         */
    ETA INTEGER(10),S=ASCII,R=DEC,F=' ' /*ESTIMATED TIME OF ARRIVAL          */
    SOA STRING(4)        /*INTENDED SPEED OF ADVANCE,VALUES FROM 0.0 TO 99.9*/
    ESCTDESIG STRING(9)                   /*DESIGNATION OF THE ESCORT GROUP  */
                                          /* ASSIGNED TO THE CONVOY          */

    TRACKCOUNT INTEGER(1),S=ASCII,R=DEC /* IF 1 NEXT PART GIVES CURRENT POS*/
    PTP  STRING(12)                       /*GEOGRAPHIC COORDINATES           */
    PROB STRING(11)                       /*UNCERTAINTY AREA PARAMETERS      */
    PTC INTEGER(3),S=ASCII,R=DEC          /*COURSE,IN DEGREES CLOCKWISE      */
    PTS STRING(4)                /*SPEED IN KNOTS AND TENTHS: 00.0 TO 99.9   */
    PTD INTEGER(10),S=ASCII,R=DEC         /*DATE AND TIME                    */
        /* *** END OF POSITION INFORMATION *** */

    DRTRACKCOUNT INTEGER(1),S=ASCII,R=DEC         /*SAME FOR NEXT ESTIMATED  */
                                                  /*BY DEAD RECKONING        */
    DRPOSIT STRING(12)                            /* SAME AS PTP             */
    DRPROB STRING(11)                             /*UNCERTAINTY PARAMETERS   */

    DRETA INTEGER(10),S=ASCII,R=DEC               /* DATE AND TIME           */
        /* *** END OF DEAD RECKONING INFORMATION *** */
 END;
```

AD-A122 473 ARTIFICIAL INTELLIGENCE -- RESEARCH AND APPLICATIONS 3/3
(U) STANFORD RESEARCH INST MENLO PARK CALIF B RAPHAEL
JUN 76 DAHC04-75-C-0005

UNCLASSIFIED F/G 6/4 NL

END
DATE
FILMED
1 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
CREATE CLASSFILE FILE LIST (1,30,100)
  CLASSRECORD STRUCTURE
    CLASS STRING (15),I=D              /* CLASS NAME   */
    FLAG STRING(2),I=D  /*COUNTRY; VALUES ARE US,UR,UK,NE,SF,WG,NO,LI,AN   */
                        /*SA,VE,IT,SP,FR,CA,EG,PO,AR(SEE IN SHIP FILE)     */
    CAT STRING (3),I=D  /*CATEGORY; VALUES ARE:MER(CHANT),FSH(FISHING)     */
                        /*      NAV(AL)                                    */
        /*    TYPE STRUCTURE.    SHIP TYPE IN ACRONYM FORM; VALUES ARE:    */
                /*AGI(INTELLIGENCE COLLECTOR),AO(FLEET OILER)              */
                /*AOE(OILER AND AMMUNITION SHIP),BULK(CARGO FREIGHTER)     */
                /*CG(GUIDED MISS. CRUISER),CLG(GUIDED MISS. LIGHT CRUISER) */
                /*CLGN(NUCLEAR POWERED GUIDED MISSILE LIGHT CRUISER        */
                /*CV(AIRCRAFT CARRIER),DD(DESTROYER),DDG(GUIDED MISSILE    */
                /*DESTROYER),FF(FRIGATE),FFG(GUIDED MISSILE FRIGATE)       */
                /*SS(ATTACK SUBMARINE),SSN(NUCLEAR ATTACK SUBMARINE)       */
                /*SSBN(NUCLEAR POWERED BALISTIC MISSILE SUBMARINE)         */
                /*SSGN(NUCLEAR POWERED GUIDED MISSILE SUBMARINE)           */
   TYPE1 STRING (1),I=D  /*VALUES ARE:A(INTELLIGENCE COLLECTOR OR OILER)   */
                         /*B(BULK),C(CRUISER OR CARRIER),D(DESTROYER)      */
                         /*F(FRIGATE),S(SUB)                               */
   TYPE2 STRING (1),I=D  /*VALUES ARE:D(ESTROYER),F(RIGATE)                */
                         /*G(INT. COL. OR GUIDED MISSILE),L(LIGHT CRUISER) */
                         /*O(OILER),S(SUB),U(BULK),V(AIRCRAFT CARRIER)     */
   TYPE3 STRING (1),I=D  /* VALUES ARE:B(SSBN),E(AOE)                      */
                         /*G(GUIDED MISSILE),L(BULK),N(UCLEAR)             */
   TYPE4 STRING (1),I=D  /* VALUES ARE:K(BULK),N(NUCLEAR)                  */
   TYPE STRING(4),VE=TYPE1!TYPE2!TYPE3!TYPE4,I=D

   LGH INTEGER (4),S=ASCII,R=DEC,F=' '           /*LENGTH IN FEET          */
   BEAM INTEGER(3),S=ASCII,R=DEC                 /*WIDTH IN FEET           */
   DRAFT INTEGER (2),S=ASCII,R=DEC               /*DRAFT IN FEET           */
        /*    FTP STRUCTURE             FUEL TYPE CODE */
   FTP1 STRING(1)    /*FIRST CHAR. IS A NUMBER WHICH INDICATES ENGINE TYPE, */
                     /*VALUES ARE:1(STEAM TURBINE),2(GAS TURB.),3(DIESEL)   */
   FTP2 STRING(1)    /* 2ND CHAR. IS A LETTER; VALUES ARE:A(BUNKER A)       */
                     /*B(BUNKER B),C(BUNKER C),D(IESEL),J(DISTILLATE OR JP-5)*/
                     /*N(UCLEAR)                                            */
   FTP STRING(2),VE=FTP1!FTP2

   MCS STRING(4)          /* MAXIMUM CRUISING SPEED IN KNOTS AND TENTHS     */
   MCM STRING(5)          /*MAX. CRUISING RANGE IN NAUTICAL MILES AT MAXIMUM */
                          /*SPEED; FOR NUCLEAR, THIS FIELD IS EMPTY         */
   NCS STRING (4)         /* NORMAL(ECONOMICAL)CRUISING SPEED               */
   NCM STRING (5)         /*NORMAL CRUISING RANGE(MCM,NCS,NCM) ARE EMPTY    */
                          /*FOR NUCLEAR SUBS)                               */

        /* *** END OF PART COMMON TO ALMOST ALL SHIPS *** */
```

```
MERFLG INTEGER(1),S=ASCII,R=DEC /*1 FOR MERCHANT SHIP, 0 O/WISE    */
  MERSUBPART LIST(0,1),C=MERFLG         /*NEXT PART FOR MERCHANT SHIPS ONLY*/
    MERSUBRECORD STRUCTURE
      NAT STRING(2),I=I          /*NATIONALITY OF SHIP REGISTRATION          */
      OWN STRING(2),I=I          /*NATIONALITY OF THE OWNERS OF THE SHIP     */
      DWT INTEGER(6),S=ASCII,R=DEC       /*DEAD WEIGHT IN TONS              */
      GWT INTEGER (7),S=ASCII,R=DEC      /*GROSS REGISTERED TONNAGE         */
    END         /* *** END OF PART FOR MERCHANT SHIPS *** */

  NAVALFLG INTEGER(1),S=ASCII,R=DEC      /* 1 IF NAVAL SHIP, 0 O/WISE        */
  NAVALPART LIST(0,1),C=NAVALFLG         /* NEXT PART FOR NAVAL SHIPS ONLY   */
    NAVALRECORD STRUCTURE
      DISPL INTEGER(5),S=ASCII,R=DEC  /*DISPLACEMENT IN TONS                 */
      ENDUR INTEGER(3),S=ASCII,R=DEC  /*CRUISING ENDURANCE IN DAYS BASED ON*/
                                      /*STAPLES CARRIED AND CREW FATIGUE     */

      GUNFLG INTEGER(1),S=ASCII,R=DEC           /*   OF GUN SIZES ON BOARD  */
      GUNPART LIST(0,1,9),C=GUNFLG
        GUNRECORD STRUCTURE               /* ONE RECORD PER GUN SIZE         */
          GUNSIZE STRING(7)               /* GUNBORE DIAMETER,CALIBER AND    */
                                          /*UNIT OF MEASURE; EXAMPLE:5"/54   */
                                          /*INDICATES A 5 INCH 54 CALIBER GUN*/
          GUNS INTEGER(2),S=ASCII,R=DEC /*  OF GUNS                          */
        END            /* *** END OF GUN RECORD *** */

      ASLNCH STRING(8)                    /*ASW LAUNCHER TYPE DESIGNATION    */
      LNCHRS INTEGER(2),S=ASCII,R=DEC   /*   OF ASW LAUNCHERS                */
      TNOMFLG INTEGER(1),S=ASCII,R=DEC  /*   OF TORPEDO SIZES                */
      TNOMPART LIST(0,1,9),S=ASCII,R=DEC
        TNOMRECORD STRUCTURE              /* 1 RECORD PER TORPEDO SIZE       */
          TNOM STRING(4)                   /* TORPEDO TUBE TYPE DESIGNATION  */
          TSIZE INTEGER(2),S=ASCII,R=DEC /* TORPEDO TUBE DIAMETER IN INCHES*/
          TUBES INTEGER(2),S=ASCII,R=DEC /*   OF TORPEDO TUBES ON BOARD     */
        END            /* *** END OF TORPEDO RECORD *** */

      MISSLFLG INTEGER(1),S=ASCII,R=DEC /*   OF MISSILE TYPES ON BOARD      */
      MISSLPART LIST(0,1,5),C=MISSLFLG
        MISSLRECORD STRUCTURE
          MISSL STRING(12)                        /*DESIG. OR NAME OF MISSILE*/
          MISLNCH INTEGER(2),S=ASCII,R=DEC      /*  OF LAUNCHERS ON BOARD  */
          MISRNG INTEGER(4),S=ASCII,R=DEC /*MISSILE RANGE IN NAUTICAL MILES*/
        END   /* *** END OF MISSILE RECORD *** */
    END                 /* *** END OF NAVAL SHIPS PART *** */
  END;


CREATE PORTFILE FILE LIST(1,50,200)
  PORTRECORD STRUCTURE
    DEP STRING (1,10,40),D=' ',I=D       /* PORT NAME                      */
    DPC STRING (2),I=D                   /* COUNTRY                        */
    PTP STRING (4,10,12),D=' '           /* GEOGRAPHIC COORDINATES         */
  END;
```

Appendix IV

LISTING OF DATA IN COMMAND AND CONTROL DATA BASE

APPENDIX IV

LISTING OF DATA IN COMMAND AND CONTROL DATA BASE

SHIPFILE: from NAME to PCFUEL

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
MINSK	KURIL	UR	NAV	CV	4	RN01	*	*	083
KIEV	KURIL	UR	NAV	CV	3	RN02	*	*	074
MOSKVA	MOSKVA	UR	NAV	CV	1	RN03	*	*	088
LENINGRAD	MOSKVA	UR	NAV	CV	2	RN04	*	*	092
*	DELTA	UR	NAV	SSBN	901	RN05	*	*	100
*	DELTA	UR	NAV	SSBN	902	RN06	*	*	100
*	YANKEE	UR	NAV	SSBN	501	RN07	*	*	100
*	YANKEE	UR	NAV	SSBN	502	RN08	*	*	100
*	YANKEE	UR	NAV	SSBN	518	RN09	*	*	100
*	ECHO II	UR	NAV	SSGN	301	RN10	*	*	100
*	ECHO II	UR	NAV	SSGN	307	RN11	*	*	100
*	ECHO II	UR	NAV	SSGN	312	RN12	*	*	100
*	ECHO II	UR	NAV	SSGN	337	RN13	*	*	100
AMPERMETR	OKEAN	UR	NAV	AGI	128	RN14	*	*	054
BAROGRAPH	OKEAN	UR	NAV	AGI	211	RN15	*	*	077
KRENOMETR	OKEAN	UR	NAV	AGI	174	RN16	*	*	044
*	CHARLIE	UR	NAV	SSGN	551	RN17	*	*	100
*	CHARLIE	UR	NAV	SSGN	552	RN18	*	*	100
*	CHARLIE	UR	NAV	SSGN	557	RN19	*	*	100
*	CHARLIE	UR	NAV	SSGN	564	RN20	*	*	100
*	CHARLIE	UR	NAV	SSGN	566	RN21	*	*	100
*	CHARLIE	UR	NAV	SSGN	570	RN22	*	*	100
*	CHARLIE	UR	NAV	SSGN	571	RN23	*	*	100
*	CHARLIE	UR	NAV	SSGN	574	RN24	*	*	100
ADMIRAL FOKIN	KYNDA	UR	NAV	CLG	854	RN25	*	*	068
ADMIRAL GOLOVKO	KYNDA	UR	NAV	CLG	855	RN26	*	*	072
GROZNY	KYNDA	UR	NAV	CLG	856	RN27	*	*	054
VARYAG	KYNDA	UR	NAV	CLG	857	RN28	*	*	063
SVERDLOV	SVERDLOV	UR	NAV	CA	840	RN29	*	*	094
ALEKSANDR SUVOROV	SVERDLOV	UR	NAV	CA	841	RN30	*	*	087
MURMANSK	SVERDLOV	UR	NAV	CA	842	RN31	*	*	089
DMITRI POZHARSKI	SVERDLOV	UR	NAV	CA	843	RN32	*	*	065
MIKHAIL KUTUSOV	SVERDLOV	UR	NAV	CA	844	RN33	*	*	044
ADMIRAL ISAKOV	KRESTA II	UR	NAV	CLG	580	RN34	*	*	056
ADMIRAL MAKAROV	KRESTA II	UR	NAV	CLG	581	RN35	*	*	088
KRONSTADT	KRESTA II	UR	NAV	CLG	582	RN36	*	*	084
ALATYR	KAZBEK	UR	NAV	AO	132	RN37	*	*	099

SHIPFILE: from NAME to PCFUEL (continued)

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
DESNA	KAZBEK	UR	NAV	AO	133	RN38	*	*	090
ANDREY	KAZBEK	UR	NAV	AO	134	RN39	*	*	040
VOLKHOV	KAZBEK	UR	NAV	AO	135	RN40	*	*	067
OBRAZSTSOVY	KASHIN	UR	NAV	DDG	564	RN41	*	*	055
PROVORNY	KASHIN	UR	NAV	DDG	565	RN42	*	*	040
SKORY	KASHIN	UR	NAV	DDG	570	RN43	*	*	075
SLAVNY	KASHIN	UR	NAV	DDG	225	RN44	*	*	070
OTVAZHNY	KASHIN	UR	NAV	DDG	197	RN45	*	*	050
GREENVILLE VICTORY	VICTORY	US	MER	BULK	*	UA1A	*	CTW09	100
JOHN TOULE	VICTORY	US	MER	BULK	*	UA1B	D	CTW09	100
FRANCIS MCGRAW	VICTORY	US	MER	BULK	*	UA1C	D	CTW09	100
ANDREW MILLER	VICTORY	US	MER	BULK	*	UA1D	*	CTW09	100
MORRIS E CRAIN	VICTORY	US	MER	BULK	*	UA1E	*	CTW09	100
TRUMAN KIMLOW	VICTORY	US	MER	BULK	*	UA1F	*	CTW09	100
JAMES E.ROBINSON	VICTORY	US	MER	BULK	*	UA1G	D	CTW09	100
JOSEPH E.MERRILL	VICTORY	US	MER	BULK	*	UA1H	*	CTW09	100
JACK J.PENDLETON	VICTORY	US	MER	BULK	*	UA1I	*	CTW09	100
PACIFIC	SEALIFT	US	MER	TNKR	*	UA1J	*	AN72	100
ATLANTIC	SEALIFT	US	MER	TNKR	*	UA1K	*	AN72	100
ARABIAN SEA	SEALIFT	US	MER	TNKR	*	UA1L	*	AN72	100
MEDITERRANEAN	SEALIFT	US	MER	TNKR	*	UA1M	*	AN72	100
CHINA SEA	SEALIFT	US	MER	TNKR	*	UA1N	*	AN72	100
CARRIBEAN	SEALIFT	US	MER	TNKR	*	UA1O	*	AN72	100
INDIAN OCEAN	SEALIFT	US	MER	TNKR	*	UA1P	*	AN72	100
ARCTIC	SEALIFT	US	MER	TNKR	*	UA1Q	*	AN72	100
ANTARCTIC	SEALIFT	US	MER	TNKR	*	UA1R	*	AN72	100
SUAMICO	MISSION	US	MER	TNKR	*	UA1T	*	AN72	100
TALLULAH	MISSION	US	MER	TNKR	*	UA1U	*	AN72	100
PECOS	MISSION	US	MER	TNKR	*	UA1V	*	AN72	100
MILLICOMA	MISSION	US	MER	TNKR	*	UA1W	*	AN72	100
SAUGATUCK	MISSION	US	MER	TNKR	*	UA1X	*	AN72	100
SCHUYLKILL	MISSION	US	MER	TNKR	*	UA1Y	*	AN72	100
COSSATOT	MISSION	US	MER	TNKR	*	UA1Z	*	AN72	100
SANTA INEZ	MISSION	US	MER	TNKR	*	UA2A	*	AN72	100
ADELAIDE STAR	BLUESTAR	UK	MER	BULK	*	KU7A	D	NL53	100
AMERICA STAR	BLUESTAR	UK	MER	BULK	*	KU7B	*	NL53	100
ARGENTINA STAR	BLUESTAR	UK	MER	BULK	*	KU7C	*	NL53	100
AUSTRALIA STAR	BLUESTAR	UK	MER	BULK	*	KU7D	*	NL53	100
BRASIL STAR	BLUESTAR	UK	MER	BULK	*	KU7E	*	NL53	100
CALEDONIA STAR	BLUESTAR	UK	MER	BULK	*	KU7F	D	NL53	100
CALIFORNIA STAR	BLUESTAR	UK	MER	BULK	*	KU7G	*	NL53	100
CANADIAN STAR	BLUESTAR	UK	MER	BULK	*	KU7H	*	NL53	100
CANTERBURY STAR	BLUESTAR	UK	MER	BULK	*	KU7J	*	NL53	100
COLORADO STAR	BLUESTAR	UK	MER	BULK	*	KU7I	*	NL53	100

SHIPFILE: from NAME to PCFUEL (continued)

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
DUNEDIN STAR	BLUESTAR	UK	MER	BULK	*	KU7K	*	NL53	100
EMPIRE STAR	BLUESTAR	UK	MER	BULK	*	KU7L	*	NL53	100
ENGLISH STAR	BLUESTAR	UK	MER	BULK	*	KU7M	*	NL53	100
ENDEAVOUR	ENDEAVOUR	UK	MER	TNKR	*	KU7N	*	NL53	100
ENTERPRISE	ENDEAVOUR	UK	MER	TNKR	*	KU7O	*	NL53	100
BRITISH ADVENTURE	ENDEAVOUR	UK	MER	TNKR	*	KU7P	*	NL53	100
BRITISH BOMBARDIER	ENDEAVOUR	UK	MER	TNKR	*	KU7Q	*	NL53	100
BRITISH BULLDOG	ENDEAVOUR	UK	MER	TNKR	*	KU7R	*	NL53	100
BRITISH CAPTAIN	ENDEAVOUR	UK	MER	TNKR	*	KU7S	*	NL53	100
BRITISH CAVALIER	ENDEAVOUR	UK	MER	TNKR	*	KU7T	*	NL53	100
ALINDA	ALINDA	UK	MER	TNKR	*	KU7U	*	NL53	100
HADRA	ALINDA	UK	MER	TNKR	*	KU7V	*	NL53	100
HADRIANINA	ALINDA	UK	MER	TNKR	*	KU7W	*	NL53	100
HANNINEA	ALINDA	UK	MER	TNKR	*	KU7X	*	NL53	100
HARPULA	ALINDA	UK	MER	TNKR	*	KU7Y	*	NL53	100
HASTULA	ALINDA	UK	MER	TNKR	*	KU7Z	*	NL53	100
HATASIA	ALINDA	UK	MER	TNKR	*	KU8A	*	NL53	100
HANSTELLUM	ALINDA	UK	MER	TNKR	*	KU8B	*	NL53	100
HANSTRUM	ALINDA	UK	MER	TNKR	*	KU8C	*	NL53	100
HELCION	ALINDA	UK	MER	TNKR	*	KU8D	*	NL53	100
HELDIA	ALINDA	UK	MER	TNKR	*	KU8E	*	NL53	100
HELISOMA	ALINDA	UK	MER	TNKR	*	KU8F	*	NL53	100
CONSTANTINE	NIARCHOS	LI	MER	TNKR	*	Q3A1	*	AN72	100
EUGENIE	NIARCHOS	LI	MER	TNKR	*	Q3A2	*	AN72	100
SPYROS	NIARCHOS	LI	MER	TNKR	*	Q3A3	*	AN72	100
NORTHERN JOY	NIARCHOS	LI	MER	TNKR	*	Q3A4	*	AN72	100
WORLD BOND	NIARCHOS	LI	MER	TNKR	*	Q3A5	*	AN72	100
AMSTELDIEP	AMSTERDAM	NE	MER	BULK	*	P4R3	*	NL53	100
AMSTELHOCK	AMSTERDAM	NE	MER	BULK	*	P4R4	*	NL53	100
AMSTELHOF	AMSTERDAM	NE	MER	BULK	*	P4R5	*	NL53	100
AMSTELKROON	AMSTERDAM	NE	MER	BULK	*	P4R6	*	NL53	100
AMSTELMEER	AMSTERDAM	NE	MER	BULK	*	P4R7	*	NL53	100
AMSTELMOLEN	AMSTERDAM	NE	MER	BULK	*	P4R8	*	NL53	100
AMSTELSLUIS	AMSTERDAM	NE	MER	BULK	*	P4R9	*	NL53	100
AMSTELSTAD	AMSTERDAM	NE	MER	BULK	*	P4S0	*	NL53	100
AMSTELVEEN	AMSTERDAM	NE	MER	BULK	*	P4S1	*	NL53	100
AMSTELVELD	AMSTERDAM	NE	MER	BULK	*	P4S2	*	NL53	100
MERCHANT	SPRINGBOK	SF	MER	BULK	*	P3A2	*	CTW09	100
PIONEER	SPRINGBOK	SF	MER	BULK	*	P3A3	*	CTW09	100
SEAFARER	SPRINGBOK	SF	MER	BULK	*	P3A4	*	CTW09	100
SHIPPER	SPRINGBOK	SF	MER	BULK	*	P3A5	*	CTW09	100
STATESMAN	SPRINGBOK	SF	MER	BULK	*	P3A6	*	CTW09	100
TRADER	SPRINGBOK	SF	MER	BULK	*	P3A7	*	CTW09	100
TRANSPORTER	SPRINGBOK	SF	MER	BULK	*	P3A8	*	CTW09	100

SHIPFILE: from NAME to PCFUEL (continued)

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
VANGUARD	SPRINGBOK	SF	MER	BULK	*	P3A9	*	CTW09	100
VENTURE	SPRINGBOK	SF	MER	BULK	*	P3B0	*	CTW09	100
VICTORY	SPRINGBOK	SF	MER	BULK	*	P3B1	*	CTW09	100
POSEIDON	STINNES	WG	MER	BULK	*	T5A3	*	NL53	100
TRANSAMERICA	STINNES	WG	MER	BULK	*	T5A4	*	NL53	100
TRANSATLANTIC	STINNES	WG	MER	BULK	*	T5A5	*	NL53	100
TRANSCANADA	STINNES	WG	MER	BULK	*	T5A6	*	NL53	100
TRANSEUROPA	STINNES	WG	MER	BULK	*	T5A7	*	NL53	100
TRANSGERMANIA	STINNES	WG	MER	BULK	*	T5A8	*	NL53	100
TRANSPACIFIC	STINNES	WG	MER	BULK	*	T5A9	*	NL53	100
TRANSQUEBEC	STINNES	WG	MER	BULK	*	T5B0	*	NL53	100
TABOR	WILHELMSON	NO	MER	BULK	*	K4P0	*	*	100
TAGAYTRAY	WILHELMSON	NO	MER	BULK	*	K4P1	*	*	100
TAGRIS	WILHELMSON	NO	MER	BULK	*	K4P2	*	*	100
TAIPING	WILHELMSON	NO	MER	BULK	*	K4P3	*	*	100
TALABOT	WILHELMSON	NO	MER	BULK	*	K4P4	*	*	100
TALISMAN	WILHELMSON	NO	MER	BULK	*	K4P5	D	*	100
TALLEYRAND	WILHELMSON	NO	MER	BULK	*	K4P6	*	*	100
TAMESIS	WILHELMSON	NO	MER	BULK	*	K4P7	*	*	100
TAMPA	WILHELMSON	NO	MER	BULK	*	K4P8	D	*	100
TANA	WILHELMSON	NO	MER	BULK	*	K4P9	*	*	100
TANCRED	WILHELMSON	NO	MER	BULK	*	K4Q0	*	*	100
TARANTED	WILHELMSON	NO	MER	BULK	*	K4Q1	*	*	100
TARIFA	WILHELMSON	NO	MER	BULK	*	K4Q2	*	*	100

SHIPFILE: from NAME to PCFUEL (continued)

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
TARU	WILHELMSON	NO	MER	BULK	*	K4Q3	*	*	100
TASCO	WILHELMSON	NO	MER	BULK	*	K4Q4	*	*	100
TAURUS	WILHELMSON	NO	MER	BULK	*	K4Q5	*	*	100
TERNA	WILHELMSON	NO	MER	BULK	*	K4Q6	*	*	100
TENNERAIRE	WILHELMSON	NO	MER	BULK	*	K4Q7	*	*	100
TENERIFFA	WILHELMSON	NO	MER	BULK	*	K4Q8	*	*	100
TENNESSEE	WILHELMSON	NO	MER	BULK	*	K4Q9	*	*	100
CONSTELLATION	KITTYHAWK	US	NAV	CV	64	NABC	D	*	089
JOHN F.KENNEDY	KITTYHAWK	US	NAV	CV	67	NABD	D	*	090
KITTYHAWK	KITTYHAWK	US	NAV	CV	63	NABE	D	*	088
AMERICA	KITTYHAWK	US	NAV	CV	66	NABF	D	*	000
SARATOGA	FORRESTAL	US	NAV	CV	60	NABG	D	*	100
INDEPENDENCE	FORRESTAL	US	NAV	CV	62	NABH	D	*	100
LOS ANGELES	LOS ANGELES	US	NAV	SSN	688	NABI	*	*	100
BATON ROUGE	LOS ANGELES	US	NAV	SSN	689	NABJ	*	*	100
PHILADELPHIA	LOS ANGELES	US	NAV	SSN	690	NABK	*	*	100
STURGEON	STURGEON	US	NAV	SSN	637	NABL	*	*	100
WHALE	STURGEON	US	NAV	SSN	638	NABM	*	*	100
TAUTOG	STURGEON	US	NAV	SSN	639	NABN	*	*	100
GRAYLING	STURGEON	US	NAV	SSN	646	NABO	*	*	100
POGY	STURGEON	US	NAV	SSN	647	NABP	*	*	100
ASPRO	STURGEON	US	NAV	SSN	648	NABQ	*	*	100
SUNFISH	STURGEON	US	NAV	SSN	649	NABR	*	*	100
CALIFORNIA	CALIFORNIA	US	NAV	CLGN	36	NABS	*	*	100
SOUTH CAROLINA	CALIFORNIA	US	NAV	CLGN	37	NABT	*	*	100
JOSEPHUS DANIELS	BELKNAP	US	NAV	CLG	27	NABU	*	*	076
WAINWRIGHT	BELKNAP	US	NAV	CLG	28	NABV	D	*	088
JOUETT	BELKNAP	US	NAV	CLG	29	NABW	D	*	090
HORNE	BELKNAP	US	NAV	CLG	30	NABX	D	*	078
STERETT	BELKNAP	US	NAV	CLG	31	NABY	D	*	091
WILLIAM H.STANDLEY	BELKNAP	US	NAV	CLG	32	NABZ	D	*	098
FOX	BELKNAP	US	NAV	CLG	33	NACA	D	*	087
BIDDLE	BELKNAP	US	NAV	CLG	34	NACB	D	*	088
LEAHY	LEAHY	US	NAV	CLG	16	NACC	D	*	096
HARRY E.YARNELL	LEAHY	US	NAV	CLG	17	NACD	D	*	078
WORDEN	LEAHY	US	NAV	CLG	18	NACE	D	*	088
DALE	LEAHY	US	NAV	CLG	19	NACF	*	*	078
RICHMOND K.TURNER	LEAHY	US	NAV	CLG	20	NACG	D	*	067
GRIDLEY	LEAHY	US	NAV	CLG	21	NACH	D	*	077
ENGLAND	LEAHY	US	NAV	CLG	22	NACI	D	*	066
HALSEY	LEAHY	US	NAV	CLG	23	NACJ	*	*	085
REEVES	LEAHY	US	NAV	CLG	24	NACK	D	*	085
CHARLES F.ADAMS	CHARLES F.ADAMS	US	NAV	DDG	2	NACL	*	AN72	080
JOHN KING	CHARLES F.ADAMS	US	NAV	DDG	3	NACM	D	AN72	080
LAWRENCE	CHARLES F.ADAMS	US	NAV	DDG	4	NACN	*	AN72	086
CLAUDE V.RICKETTS	CHARLES F.ADAMS	US	NAV	DDG	5	NACO	*	AN72	083

SHIPFILE: from NAME to PCFUEL (continued)

NAME	CLASS	FLAG	CAT	TYPE	HUL	IRCS	DOCTR	TITLE	PCFUEL
BARNEY	CHARLES F.ADAMS	US	NAV	DDG	6	NACP	*	AN72	095
HENRY B.WILSON	CHARLES F.ADAMS	US	NAV	DDG	7	NACQ	*	AN72	090
LYNDE B.MCCORMICK	CHARLES F.ADAMS	US	NAV	DDG	8	NACR	*	C2A2	095
TOWERS	CHARLES F.ADAMS	US	NAV	DDG	9	NACS	*	C2A2	090
SELLERS	CHARLES F.ADAMS	US	NAV	DDG	11	NACT	*	C2A2	090
ROBISON	CHARLES F.ADAMS	US	NAV	DDG	12	NACU	*	C2A2	089
HOEL	CHARLES F.ADAMS	US	NAV	DDG	13	NACV	*	C2A2	079
KNOX	KNOX	US	NAV	FF	1052	NACW	*	C3Z6	090
ROARK	KNOX	US	NAV	FF	1053	NACX	*	C3Z6	088
GRAY	KNOX	US	NAV	FF	1054	NACY	*	C3Z6	090
HEPBURN	KNOX	US	NAV	FF	1055	NACZ	*	C3Z6	079
CONNOLE	KNOX	US	NAV	FF	1056	NADA	*	C3Z6	078
RATHBURNE	KNOX	US	NAV	FF	1057	NADB	*	C3Z6	067
MEYERKORD	KNOX	US	NAV	FF	1058	NADC	D	C3Z6	088
W.S.SIMS	KNOX	US	NAV	FF	1059	NADD	*	C3Z6	079
LANG	KNOX	US	NAV	FF	1060	NADE	*	C3Z6	088
HASSAYAMPA	HASSAYAMPA	US	NAV	AO	148	NADF	*	*	098
KAWISHIWI	HASSAYAMPA	US	NAV	AO	150	NADG	*	*	087
ASHTABULA	HASSAYAMPA	US	NAV	AO	151	NADH	*	*	075

SHIPFILE: from USNAVALFLG to OPCON

NAME	USNAV FLG	HOGEO	CONAM	LINEAL	OPCON
MINSK	0				
.	.				
.	.				
.	.				
TENNESSEE	0				
CONSTELLATION	1	MAYP	CAPT J.ELLISON	000004832	CTG67.2
JOHN F.KENNEDY	1	MAYP	CAPT P.MOFFETT	000004833	CTG27.2
KITTYHAWK	1	MAYP	CAPT R.SPRUANCE	000004834	CTG67.1
AMERICA	1	NORF	CAPT W.HALSEY	000004835	CTG67.3
SARATOGA	1	NORF	CAPT A.BROWN	000004836	CTG67.3
INDEPENDENCE	1	MAYP	CAPT S.JACKSON	000004837	CTG67.3
LOS ANGELES	1	NORF	CDR D.JONES	000004838	COMSUBLANT
BATON ROUGE	1	NORF	CDR V.QUIET	000004839	COMSUBLANT
PHILADELPHIA	1	NORF	CDR L.SNEAK	000004840	COMSUBLANT
STURGEON	1	NORF	CDR R.SMITH	000010100	COMSUBLANT
WHALE	1	NORF	CDR X.COHEN	000010101	COMSUBLANT
TAUTOG	1	NORF	CDR J.HIGH	000010102	COMSUBLANT
GRAYLING	1	NORF	CDR R.DAUGHERTY	000010103	COMSUBLANT
POGY	1	NORF	CDR J.HORNER	000010104	COMSUBLANT
ASPRO	1	NORF	CDR T.CHANDLER	000010105	COMSUBLANT
SUNFISH	1	NORF	CDR M.MORTON	000010106	COMSUBLANT
CALIFORNIA	1	CHAR	CAPT R.MORRIS	000004841	CTU67.2.1
SOUTH CAROLINA	1	CHAR	CAPT J.KEELY	000004842	CTU27.7.1
JOSEPHUS DANIELS	1	CHAR	CAPT J.HARMS	000004843	CTU67.1.1
WAINWRIGHT	1	CHAR	CAPT O.EVANS	000004844	CTU67.1.1
JOUETT	1	CHAR	CAPT T.FRENZINGER	000004845	CTU67.1.1
HORNE	1	CHAR	CAPT J.BRANIN	000004846	CTU67.1.1
STERETT	1	CHAR	CAPT W.HOHMANN	000004847	CTU67.1.1
WILLIAM H.STANDLEY	1	CHAR	CAPT C.MICHAELS	000004848	CTU67.1.1
FOX	1	CHAR	CAPT J.EVERETT	000004849	CTU67.2.1
BIDDLE	1	CHAR	CAPT J.TOWNES	000004850	CTU67.2.1
LEAHY	1	CHAR	CAPT H.GRAHAM	000004851	CTU67.2.1
HARRY E.YARNELL	1	CHAR	CAPT P.PHILHOWER	000004852	CTU67.2.1
WORDEN	1	CHAR	CAPT J.YOUNG	000004853	CTU67.2.1
DALE	1	CHAR	CAPT R.GRANTHAM	000004854	CTU27.7.1
RICHMOND K.TURNER	1	CHAR	CAPT C.WAHL	000004855	CTU27.7.1
GRIDLEY	1	CHAR	CAPT G.BROWN	000004856	CTU27.7.1

SHIPFILE: from USNAVALFLG to OPCON (continued)

NAME	USNAV FLG	HOGEO	CONAM	LINEAL	OPCON
ENGLAND	1	CHAR	CAPT R.SMITH	000004857	CTU27.7.1
HALSEY	1	CHAR	CAPT J.HOBLITZELL	000004858	CTU27.7.1
REEVES	1	CHAR	CAPT R.HOOLHORST	000004859	CTU27.7.1
CHARLES F.ADAMS	1	NORF	CDR W.BURNS	000010001	CTU24.2.1
JOHN KING	1	NORF	CDR J.P.JONES	000010002	CTU24.2.1
LAWRENCE	1	NORF	CDR R.BRANDENBURG	000010003	CTU24.2.1
CLAUDE V.RICKETTS	1	NORF	CDR F.HOLLISTER	000010004	CTU24.2.1
BARNEY	1	NORF	CDR J.FOXX	000010005	CTU24.2.1
HENRY B.WILSON	1	NORF	CDR W.T.DOOR	000010006	CTU24.2.1
LYNDE B.MCCORMICK	1	NORF	CDR W.T.HATCH	000010007	CTU24.2.2
TOWERS	1	NORF	CDR P.OSGOOD	000010008	CTU24.2.2
SELLERS	1	NORF	CDR C.PRESGROVE	000010009	CTU24.2.2
ROBISON	1	NORF	CDR A.BURKE	000010011	CTU24.2.2
HOEL	1	NORF	CDR W.HUNT	000010010	CTU24.2.2
KNOX	1	CHAR	CDR C.JACKSON	000010012	CTU24.2.2
ROARK	1	CHAR	CDR J.ELLIOTT	000010013	CTU24.2.3
GRAY	1	CHAR	CDR P.LILLY	000010014	CTU24.2.3
HEPBURN	1	CHAR	CDR D.WEISGERBER	000010015	CTU24.2.3
CONNOLE	1	CHAR	CDR W.CARL	000010016	CTU24.2.3
RATHBURNE	1	CHAR	CDR W.MORAN	000010017	CTU24.2.3
MEYERKORD	1	CHAR	CDR P.RILEY	000010018	CTU24.2.3
W.S.SIMS	1	CHAR	CDR D.RODGERS	000010019	CTU24.2.3
LANG	1	CHAR	CDR D.LEACH	000010020	CTU24.2.3
HASSAYAMPA	1	NORF	CAPT R.SHELL	000004860	CTG67.2
KAWISHIWI	1	NORF	CAPT T.MOBIL	000004861	CTG27.7
ASHTABULA	1	NORF	CAPT A.ARCO	000004862	CTG67.1

SHIPFILE: from READY to EEND

NAME	USNAV FLG	READY	REASN	EIC	CASREP		CADAT	ETERM	EBEG	EEND
MINSK	0									
.	.									
.	.									
.	.									
TENERIFFA	0									
TENNESSEE	0									
CONSTELLATION	1	1	*	*	0000000000	0	000000	SURVOPS	760110	760228
JOHN F.KENNEDY	1	1	*	*	0000000000	0	000000	SURVOPS	751215	760215
KITTYHAWK	1	2	A	*	7601162330	1	760119	SURVOPS	760103	760205
AMERICA	1	5	O	*	0000000000	1	760601	OVHL	760101	760601
SARATOGA	1	1	*	*	0000000000	0	000000	RAV	760115	760120
INDEPENDENCE	1	1	*	*	0000000000	0	000000	RAV	760114	760119
LOS ANGELES	1	1	*	*	0000000000	0	000000	ASOPS	760101	760301
BATON ROUGE	1	1	*	*	0000000000	0	000000	ASOPS	751215	760315
PHILADELPHIA	1	1	*	*	0000000000	0	000000	ASOPS	760115	760415
STURGEON	1	1	*	*	0000000000	0	000000	UPKEEP	760101	760301
WHALE	1	1	*	*	0000000000	0	000000	UPKEEP	760131	760601
TAUTOG	1	1	*	*	0000000000	0	000000	UPKEEP	760115	760130
GRAYLING	1	1	*	*	0000000000	0	000000	UPKEEP	760101	760131
POGY	1	1	*	*	0000000000	0	000000	ANTSHIP	751115	760212
ASPRO	1	1	*	*	0000000000	0	000000	SURVOPS	751215	760315
SUNFISH	1	1	*	*	0000000000	0	000000	SURVOPS	751201	760301
CALIFORNIA	1	1	*	*	0000000000	0	000000	CARESC	760101	760601
SOUTH CAROLINA	1	1	*	*	0000000000	0	000000	CARESC	760115	760301
JOSEPHUS DANIELS	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
WAINWRIGHT	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
JOUETT	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
HORNE	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
STERETT	1	3	S	*	7601162230	1	760118	CARESC	751231	760615
WILLIAM H.STANDLEY	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
FOX	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
BIDDLE	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
LEAHY	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
HARRY E.YARNELL	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
WORDEN	1	1	*	*	0000000000	0	000000	CARESC	751231	760615
DALE	1	1	*	*	0000000000	0	000000	CARESC	751215	760301

NAME	USNAV FLG	READY	REASN	EIC	CASREP		CADAT	ETERM	EBEG	EEND
RICHMOND K.TURNER	1	1	*	*	0000000000	0	000000	CARESC	751215	760301
GRIDLEY	1	1	*	*	0000000000	0	000000	CARESC	751215	760301
ENGLAND	1	1	*	*	0000000000	0	000000	CARESC	751215	760301
HALSEY	1	1	*	*	0000000000	0	000000	CARESC	751215	760301
REEVES	1	3	A	*	7601161632	1	760118	CARESC	751215	760301
CHARLES F.ADAMS	1	3	G	*	7601162330	1	760118	CONVESC	751222	760131
JOHN KING	1	1	*	*	0000000000	0	000000	CONVESC	751222	760131
LAWRENCE	1	1	*	*	0000000000	0	000000	CONVESC	751222	760131
CLAUDE V.RICKETTS	1	1	*	*	0000000000	0	000000	CONVESC	751222	760131
BARNEY	1	1	*	*	0000000000	0	000000	CONVESC	751222	760131
HENRY B.WILSON	1	1	*	*	0000000000	0	000000	CONVESC	751222	760131
LYNDE B.MCCORMICK	1	1	*	*	0000000000	0	000000	CONVESC	760112	760120
TOWERS	1	1	*	*	0000000000	0	000000	CONVESC	760112	760120
SELLERS	1	1	*	*	0000000000	0	000000	CONVESC	760112	760120
ROBISON	1	1	*	*	0000000000	0	000000	CONVESC	760112	760120
HOEL	1	2	G	*	7601141300	1	760119	CONVESC	760112	760120
KNOX	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
ROARK	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
GRAY	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
HEPBURN	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
CONNOLE	1	3	S	*	7601161600	1	760120	CONVESC	760107	760124
RATHBURNE	1	3	S	*	7601152345	1	760119	CONVESC	760107	760124
MEYERKORD	1	3	S	*	7601170030	1	760118	CONVESC	760107	760124
W.S.SIMS	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
LANG	1	1	*	*	0000000000	0	000000	CONVESC	760107	760124
HASSAYAMPA	1	1	*	*	0000000000	0	000000	REPL	760115	760125
KAWISHIWI	1	1	*	*	0000000000	0	000000	REPL	760110	760120
ASHTABULA	1	1	*	*	0000000000	0	000000	REPL	760109	760119

SHIPFILE: from MERFLG to CARGOQTY

NAME	MERFGL	HIT	PASS	CARGOTYPE	CARGOQTY
MINSK	0				
KIEV	0				
.	.				
.	.				
.	.				
SKORY	0				
SLAVNY	0				
OTVAZHNY	0				
GREENVILLE VICTORY	1	*	0012	VNAD	50
JOHN TOULE	1	*	0000	CHRORE	50
FRANCIS MCGRAW	1	*	0000	TIN	45
ANDREW MILLER	1	*	0000	TIN	49
MORRIS E CRAIN	1	*	0000	VNAD	23
TRUMAN KIMLOW	1	*	0000	VNAD	50
JAMES E. ROBINSON	1	*	0000	PHOS	37
JOSEPH E.MERRILL	1	*	0000	CHRORE	50
JACK J. PENDLETON	1	*	0000	CHRORE	50
PACIFIC	1	*	0000	OIL	350
ATLANTIC	1	*	0000	OIL	350
ARABIAN SEA	1	*	0000	OIL	350
MEDITERRANEAN	1	*	0000	OIL	350
CHINA SEA	1	*	0000	OIL	350
CARRIBEAN	1	*	0000	OIL	350
INDIAN OCEAN	1	*	0000	OIL	350
ARCTIC	1	*	0000	OIL	350
ANTARCTIC	1	*	0000	OIL	350
SUAMICO	1	*	0000	OIL	260
TALLULAH	1	*	0000	OIL	260
PECOS	1	*	0000	OIL	260
MILLICOMA	1	*	0000	OIL	260
SAUGATUCK	1	*	0000	OIL	260
SCHUYLKILL	1	*	0000	OIL	260
COSSATOT	1	*	0000	OIL	260
SANTA INEZ	1	*	0000	OIL	260
ADELAIDE STAR	1	*	0006	WHEAT	150
AMERICA STAR	1	*	0000	WHEAT	150
ARGENTINA STAR	1	*	0000	WHEAT	150
AUSTRALIA STAR	1	*	0000	ACFT	150

SHIPFILE: from MERFLG to CARGOQTY

NAME	MERFGL	HIT	PASS	CARGOTYPE	CARGOQTY
MINSK	0				
KIEV	0				
.	.				
.	.				
.	.				
SKORY	0				
SLAVNY	0				
OTVAZHNY	0				
GREENVILLE VICTORY	1	*	0012	VNAD	50
JOHN TOULE	1	*	0000	CHRORE	50
FRANCIS MCGRAW	1	*	0000	TIN	45
ANDREW MILLER	1	*	0000	TIN	49
MORRIS E CRAIN	1	*	0000	VNAD	23
TRUMAN KIMLOW	1	*	0000	VNAD	50
JAMES E. ROBINSON	1	*	0000	PHOS	37
JOSEPH E.MERRILL	1	*	0000	CHRORE	50
JACK J. PENDLETON	1	*	0000	CHRORE	50
PACIFIC	1	*	0000	OIL	350
ATLANTIC	1	*	0000	OIL	350
ARABIAN SEA	1	*	0000	OIL	350
MEDITERRANEAN	1	*	0000	OIL	350
CHINA SEA	1	*	0000	OIL	350
CARRIBEAN	1	*	0000	OIL	350
INDIAN OCEAN	1	*	0000	OIL	350
ARCTIC	1	*	0000	OIL	350
ANTARCTIC	1	*	0000	OIL	350
SUAMICO	1	*	0000	OIL	260
TALLULAH	1	*	0000	OIL	260
PECOS	1	*	0000	OIL	260
MILLICOMA	1	*	0000	OIL	260
SAUGATUCK	1	*	0000	OIL	260
SCHUYLKILL	1	*	0000	OIL	260
COSSATOT	1	*	0000	OIL	260
SANTA INEZ	1	*	0000	OIL	260
ADELAIDE STAR	1	*	0006	WHEAT	150
AMERICA STAR	1	*	0000	WHEAT	150
ARGENTINA STAR	1	*	0000	WHEAT	150
AUSTRALIA STAR	1	*	0000	ACFT	150

SHIPFILE: from MERFLG to CARGOQTY (continued)

NAME	MERFGL	HIT	PASS	CARGOTYPE	CARGOQTY
BRASIL STAR	1	*	0000	AMMO	150
CAL[illegible]NIA STAR	1	*	0000	FARMAC	150
CALIFORNIA STAR	1	*	0000	TRUCK	150
CANADIAN STAR	1	*	0000	TANKS	150
CANTERBURY STAR	1	*	0008	WHEAT	150
COLORADO STAR	1	*	0000	WHEAT	150
DUNEDIN STAR	1	*	0000	COAL	150
EMPIRE STAR	1	*	0000	COAL	150
ENGLISH STAR	1	*	0000	COAL	150
ENDEAVOUR	1	*	0000	OIL	280
ENTERPRISE	1	*	0000	OIL	280
BRITISH ADVENTURE	1	*	0000	OIL	280
BRITISH BOMBARDIER	1	*	0000	OIL	280
BRITISH BULLDOG	1	*	0000	OIL	280
BRITISH CAPTAIN	1	*	0000	OIL	280
BRITISH CAVALIER	1	*	0000	OIL	280
ALINDA	1	*	0000	OIL	70
HADRA	1	*	0000	OIL	70
HADRIANINA	1	*	0000	OIL	70
HANNINEA	1	*	0000	OIL	70
HARPULA	1	*	0000	OIL	70
HASTULA	1	*	0000	OIL	70
HATASIA	1	*	0000	OIL	70
HANSTELLUM	1	*	0000	OIL	70
HANSTRUM	1	*	0000	OIL	70
HELCION	1	*	0000	OIL	70
HELDIA	1	*	0000	OIL	70
HELISOMA	1	*	0000	OIL	70
CONSTANTINE	1	*	0000	OIL	400
EUGENIE	1	*	0000	OIL	400
SPYROS	1	*	0000	OIL	400
NORTHERN JOY	1	*	0000	OIL	400
WORLD BOND	1	*	0000	OIL	400
AMSTELDIEP	1	*	0000	COAL	180
AMSTELHOCK	1	*	0000	TRUCK	180
AMSTELHOF	1	*	0000	TANKS	180
AMSTELKROON	1	*	0000	FARMAC	180
AMSTELMEER	1	*	0000	FARMAC	180

SHIPFILE: from MERFLG to CARGOQTY (continued)

NAME	MERFGL	HIT	PASS	CARGOTYPE	CARGOQTY
AMSTELMOLEN	1	*	0000	TIN	180
AMSTELSLUIS	1	*	0000	WHEAT	180
AMSTELSTAD	1	*	0000	TUNGST	180
AMSTELVEEN	1	*	0000	TUNGST	180
AMSTELVELD	1	*	0000	TUNGST	180
MERCHANT	1	*	0000	CHRORE	150
PIONEER	1	*	0000	CHRORE	150
SEAFARER	1	*	0000	VNAD	150
SHIPPER	1	*	0000	VNAD	150
STATESMAN	1	*	0000	TUNGST	150
TRADER	1	*	0000	TUNGST	150
TRANSPORTER	1	*	0000	TUNGST	150
VANGUARD	1	*	0000	CHRORE	150
VENTURE	1	*	0000	CHRORE	150
VICTORY	1	*	0000	CHRORE	150
POSEIDON	1	*	0000	TANKS	100
TRANSAMERICA	1	*	0000	TANKS	100
TRANSATLANTIC	1	*	0000	ACFT	100
TRANSCANADA	1	*	0000	ACFT	100
TRANSEUROPA	1	*	0000	ACFT	100
TRANSGERMANIA	1	*	0000	AMMO	100
TRANSPACIFIC	1	*	0000	AMMO	100
TRANSQUEBEC	1	*	0000	AMMO	100
TABOR	1	*	0009	FOOD	150
TAGAYTRAY	1	*	0000	FOOD	150
TAGRIS	1	*	0000	FOOD	150
TAIPING	1	*	0000	FOOD	150
TALABOT	1	*	0000	FOOD	150
TALISMAN	1	*	0000	CONST	150
TALLEYRAND	1	*	0000	CONST	150
TAMESIS	1	*	0000	GENMER	150
TAMPA	1	*	0000	GENMER	150
TANA	1	*	0000	CONST	150
TANCRED	1	*	0000	FOOD	150
TARANTED	1	H	0000	TANKS	150
TARIFA	1	H	0000	ACFT	180
TARU	1	H	0000	ACFT	150
TASCO	1	H	0000	ACFT	150

SHIPFILE: from MERFLG to CARGOQTY (continued)

NAME	MERFGL	HIT	PASS	CARGOTYPE	CARGOQTY
TAURUS	1	H	0000	TANKS	150
TERNA	1	H	0000	ACFT	150
TENNERAIRE	1	H	0000	UNK	*
TENERIFFA	1	H	0000	CONST	150
TENNESSEE	1	H	0225	ACFT	125
CONSTELLATION	0				
JOHN F.KENNEDY	0				
.	.				
.	.				
.	.				

SHIPFILE: from DESTINATIONFLG to ETA

NAME	DESTIN. FLG	DEP	DPC	ETD	DST	DSC	ETA
MINSK	0	*	**	0000000000	*	**	0000000000
KIEV	0	*	**	0000000000	*	**	0000000000
.	.			.			
.	.			.			
.	.			.			
TRANSPACIFIC	0	*	**	0000000000	*	**	0000000000
TRANSQUEBEC	0	*	**	0000000000	*	**	0000000000
TABOR	1	BUENOS AIRES	AR	7601151000	OSLO	NO	7601291600
TAGAYTRAY	1	BUENOS AIRES	AR	7601151200	OSLO	NO	7601291800
TAGRIS	1	BUENOS AIRES	AR	7601140800	OSLO	NO	7601291800
TAIPING	1	BUENOS AIRES	AR	7601140800	OSLO	NO	7601292000
TALABOT	1	BALTIMORE	US	0000000000	MONROVIA	LI	7512131600
TALISMAN	1	BALTIMORE	US	0000000000	MONROVIA	LI	7512140800
TALLEYRAND	1	BALTIMORE	US	0000000000	LONDON	UK	7512010600
TAMESIS	1	LONDON	UK	0000000000	LE HAVRE	FR	7601120800
TAMPA	1	LONDON	UK	0000000000	ROTTERDAM	NE	7601141200
TANA	1	NEW YORK	US	0000000000	CARACAS	VE	7601120800
TANCRED	1	BUENOS AIRES	AR	0000000000	NAPLES	IT	7601111700
TARANTED	1	RIGA	UR	0000000000	LUANDA	AN	7601111800
TARIFA	1	NEW YORK	US	0000000000	MOCAMEDES	AN	7601220800
TARU	1	RIGA	UR	7601010030	LUANDA	AN	7601220800
TASCO	1	SEVASTOPOL	UR	7601151000	ALEXANDRIA	EG	7601181200
TAURUS	1	SEVASTOPOL	UR	7601151200	ALEXANDRIA	EG	7601181400
TERNA	1	SEVASTOPOL	UR	7601151200	ALEXANDRIA	EG	7601181500
TENNERAIRE	1	LENINGRAD	UR	0000000000	LISBON	PO	7601141600
TENERIFFA	1	LENINGRAD	UR	0000000000	LISBON	PO	7601141600
TENNESSEE	1	LENINGRAD	UR	0000000000	LISBON	PO	7601141500
CONSTELLATION	1	*	*	0000000000	NAPLES	IT	7603010800
JOHN F.KENNEDY	1	*	*	0000000000	NORFOLK	US	7603011000
KITTYHAWK	1	*	*	0000000000	NAPLES	IT	7602071200
AMERICA	1	NORFOLK	US	7608050800	*	*	0000000000
SARATOGA	1	NORFOLK	US	7601210800	6000N03000W	*	7601260800
INDEPENDENCE	1	MAYPORT	US	7601210800	3700N01700E	*	7601290800
LOS ANGELES	1	*	*	0000000000	NORFOLK	US	7604150800
BATON ROUGE	1	*	*	0000000000	NORFOLK	US	7604010800
PHILADELPHIA	1	*	*	0000000000	NORFOLK	US	7605010800
STURGEON	1	NORFOLK	US	7603020800	0000N04500E	*	7603150800
WHALE	1	NORFOLK	US	7601210800	1500S01300E	*	7601310800
TAUTOG	1	NORFOLK	US	7601310800	3700S02000E	*	7602100800
GRAYLING	1	NORFOLK	US	7602010800	3500N01000E	*	7602120800
POGY	1	3500N01000E	*	7602120800	NORFOLK	US	7602220800
ASPRO	1	*	*	0000000000	NORFOLK	US	7603181500
SUNFISH	1	*	*	0000000000	NORFOLK	US	7603060800
CALIFORNIA	0	*	**	0000000000	*	**	0000000000
.	.			.			
.	.			.			
.	.			.			
LANG	0	*	**	0000000000	*	**	0000000000
HASSAYAMPA	1	*	*	0000000000	NAPLES	IT	7601280800
KAWISHIWI	1	*	*	0000000000	NORFOLK	US	7601281000
ASHTABULA	1	*	*	0000000000	NAPLES	IT	7601221300

SHIPFILE: from TRACKCOUNT to PTD

NAME	TRACK COUNT	PTP	PROB	PTC	PTS	PTD
MINSK	1	7300N00000E	2	190	15.0	7601171200
KIEV	1	6600N00000E	5	190	15.0	7601171200
MOSKVA	1	3300N03000E	3	180	5.0	7601171200
LENINGRAD	1	3250N03010E	7	180	5.0	7601171200
*	1	7000N00700E	100/050/090	225	2.0	7601171200
*	1	6900N01200W	080/035/110	225	2.0	7601171200
*	1	4500N05200W	20	270	3.0	7601171200
*	1	3700N05900W	060/025/045	80	3.0	7601171200
*	1	2800N06700W	035/015/150	240	3.0	7601171200
*	1	5300N02000W	100/050/030	225	2.0	7601171200
*	1	4900N01700W	080/035/145	200	3.0	7601171200
*	1	4700N01500W	100/050/030	75	2.0	7601171200
*	1	4200N01700W	030/020/090	260	4.0	7601171200
AMPERMETR	1	3600N01130W	5	70	5.0	7601171200
BAROGRAPH	1	3400S01815E	5	225	4.0	7601171200
KRENOMETR	1	4330N01030W	5	275	4.0	7601171200
*	1	2310N02710W	075/025/225	80	4.0	7601171200
*	1	2150N03100W	100/050/270	90	4.0	7601171200
*	1	1945N03410W	090/060/040	75	4.0	7601171200
*	1	1940N03815W	200/100/050	100	4.0	7601171200
*	1	1630N04305W	070/020/080	95	4.0	7601171200
*	1	1615N05300W	030/015/020	105	4.0	7601171200
*	1	0900N5410E	010/005/060	110	4.0	7601171200
*	1	6000N03005W	030/005/070	70	20.0	7601171200
ADMIRAL FOKIN	1	7258N00001E	2	190	15.0	7601171200
ADMIRAL GOLOVKO	1	7259N00002W	2	190	15.0	7601171200
GROZNY	1	6558N00001E	2	180	5.0	7601171200
VARYAG	1	6559N00002W	3	180	5.0	7601171200
SVERDLOV	1	0400S04800E	5	270	10.0	7601171200
ALEKSANDR SUVOROV	1	0401S04755E	5	270	10.0	7601171200
MURMANSK	1	0402S04750E	5	270	10.0	7601171200
DMITRI POZHARSKI	1	0403S04745E	4	270	10.0	7601171200
MIKHAIL KUTUSOV	1	0404S04740E	5	270	10.0	7601171200
ADMIRAL ISAKOV	1	0401S04757E	3	270	10.0	7601171200
ADMIRAL MAKAROV	1	0402S04752E	2	270	10.0	7601171200
KRONSTADT	1	0403S04747E	5	270	10.0	7601171200
ALATYR	1	0400S04810E	6	270	10.0	7601171200
DESNA	1	0401S04810E	5	270	10.0	7601171200
ANDREY	1	0400S04812E	5	270	10.0	7601171200
VOLKHOV	1	3300N03020E	5	180	5.0	7601171200
OBRAZSTSOVY	1	4000N00610E	4	100	20.0	7601171200
PROVORNY	1	3700N01710E	5	100	20.0	7601171200
SKORY	1	0400S04758E	5	270	10.0	7601171200
SLAVNY	1	0401S04754E	5	270	10.0	7601171200
OTVAZHNY	1	0402S04749E	4	270	10.0	7601171200

SHIPFILE: from TRACKCOUNT to PTD (continued)

NAME	TRACK COUNT	PTP	PROB	PTC	PTS	PTD
GREENVILLE VICTORY	0	*	*	*	*	0000000000
JOHN TOULE	0	*	*	*	*	0000000000
.	.	.	.			.
.	.	.	.			.
.	.	.	.			.
TRANSPACIFIC	0	*	*	*	*	0000000000
TRANSQUEBEC	0	*	*	*	*	0000000000
TABOR	1	3000S04500W	0	25	16.0	7601171200
TAGAYTRAY	1	3000S04410W	0	26	15.0	7601171200
TAGRIS	1	3100S04405W	0	25	14.9	7601171200
TAIPING	1	3200S04420W	0	22	15.1	7601171200
TALABOT	0	*	*	*	*	0000000000
TALISMAN	0	*	*	*	*	0000000000
TALLEYRAND	0	*	*	*	*	0000000000
TAMESIS	0	*	*	*	*	0000000000
TAMPA	0	*	*	*	*	0000000000
TANA	0	*	*	*	*	0000000000
TANCRED	0	*	*	*	*	0000000000
TARANTED	0	*	*	*	*	0000000000
TARIFA	1	2300N030	0	130	15.0	7601171200
TARU	1	2200N02000W	5	180	16.0	7601171200
TASCO	1	3300N02800E	5	170	15.0	7601171200
TAURUS	1	3302N02800E	4	170	15.1	7601171200
TERNA	1	3303N02805E	4	180	15.0	7601171200
TENNERAIRE	0	*	*	*	*	0000000000
TENERIFFA	0	*	*	*	*	0000000000
TENNESSEE	0	*	*	*	*	0000000000
CONSTELLATION	1	4000N00600E	0	100	20.0	7601171200
JOHN F.KENNEDY	1	6000N03000W	0	70	20.0	7601171200
KITTYHAWK	1	3700N01700E	0	100	20.0	7601171200
AMERICA	0	*	*	*	*	0000000000
SARATOGA	0	*	*	*	*	0000000000
INDEPENDENCE	0	*	*	*	*	0000000000
LOS ANGELES	1	0000N04500E	0	-1	*	0760117120
BATON ROUGE	1	1500S01300E	0	-1	*	0760117120
PHILADELPHIA	1	3700S02000E	0	-1	*	0760117120
STURGEON	0	*	*	*	*	0000000000
WHALE	0	*	*	*	*	0000000000
TAUTOG	0	*	*	*	*	0000000000
GRAYLING	0	*	*	*	*	0000000000
POGY	1	3500N01000E	0	-1	*	0760117120
ASPRO	1	3000N03000W	0	180	8.0	7601171200
SUNFISH	1	3000N06000W	0	10	8.0	7601171200
CALIFORNIA	0	*	*	*	*	0000000000
SOUTH CAROLINA	0	*	*	*	*	0000000000

SHIPFILE: from DRTRACKCOUNT to DRETA

NAME	DRTRACK COUNT	DRPOSIT	DRPROB	DRETA
MINSK	0	*	*	0000000000
KIEV	0	*	*	0000000000
.	.		.	
.	.		.	
.	.		.	
LANG	0	*	*	0000000000
HASSAYAMPA	0	*	*	0000000000
KAWISHIWI	0	*	*	0000000000
ASHTABULA	0	*	*	0000000000

SHIPFILE: from DRTRACKCOUNT to DRETA

NAME	DRTRACK COUNT	DRPOSIT	DRPROB	DRETA
MINSK	0	*	*	0000000000
KIEV	0	*	*	0000000000
.	.	.		
.	.	.		
.	.	.		
LANG	0	*	*	0000000000
HASSAYAMPA	0	*	*	0000000000
KAWISHIWI	0	*	*	0000000000
ASHTABULA	0	*	*	0000000000

TRACKFILE

NAME	HUL	PTP	PROB	PTC	PTS	PTD	IND
MINSK	4	7300N00000E	2	190	15.0	7601171200	T
KIEV	3	6600N00000E	5	190	15.0	7601171200	T
MOSKVA	1	3300N03000E	3	180	5.0	7601171200	T
LENINGRAD	2	3250N03010E	7	180	5.0	7601171200	T
*	901	7000N00700E	100/050/090	225	2.0	7601171200	T
*	902	6900N01200W	080/035/110	225	2.0	7601171200	T
*	501	4500N05200W	20	270	3.0	7601171200	T
*	502	3700N05900W	060/025/045	80	3.0	7601171200	T
*	518	2800N06700W	035/015/150	240	3.0	7601171200	T
*	301	5300N02000W	100/050/030	225	2.0	7601171200	T
*	307	4900N01700W	080/035/145	200	3.0	7601171200	T
*	312	4700N01500W	100/050/030	75	2.0	7601171200	T
*	337	4200N01700W	030/020/090	260	4.0	7601171200	T
AMPERMETR	128	3600N01130W	5	70	5.0	7601171200	T
BAROGRAPH	211	3400S01815E	5	225	4.0	7601171200	T
KRENOMETR	174	4330N01030W	5	275	4.0	7601171200	T
*	551	2310N02710W	075/025/225	80	4.0	7601171200	T
*	552	2150N03100W	100/050/270	90	4.0	7601171200	T
*	557	1945N03410W	090/060/040	75	4.0	7601171200	T
*	564	1940N03815W	200/100/050	100	4.0	7601171200	T
*	566	1630N04305W	070/020/080	95	4.0	7601171200	T
*	570	1615N05300W	030/015/020	105	4.0	7601171200	T
*	571	0900N5410E	010/005/060	110	4.0	7601171200	T
*	574	6000N03005W	030/005/070	70	20.0	7601171200	T
ADMIRAL FOKIN	854	7258N00001E	2	190	15.0	7601171200	T
ADMIRAL GOLOVKO	855	7259N00002W	2	190	15.0	7601171200	T
GROZNY	856	6558N00001E	2	180	5.0	7601171200	T
VARYAG	857	6559N00002W	3	180	5.0	7601171200	T
SVERDLOV	840	0400S04800E	5	270	10.0	7601171200	T
ALEKSANDR SUVOROV	841	0401S04755E	5	270	10.0	7601171200	T
MURMANSK	842	0402S04750E	5	270	10.0	7601171200	T
DMITRI POZHARSKI	843	0403S04745E	4	270	10.0	7601171200	T
MIKHAIL KUTUSOV	844	0404S04740E	5	270	10.0	7601171200	T
ADMIRAL ISAKOV	580	0401S04757E	3	270	10.0	7601171200	T
ADMIRAL MAKAROV	581	0402S04752E	2	270	10.0	7601171200	T
KRONSTADT	582	0403S04747E	5	270	10.0	7601171200	T
ALATYR	[illegible]	0400S04810E	6	270	10.0	760117[illegible]	T
DESNA	133	0401S04810E	5	270	10.0	76011712[illegible]	[illegible]

TRACKFILE (continued)

NAME	HUL	PTP	PROB	PTC	PTS	PTD	IND
ANDREY	134	0400S04812E	5	270	10.0	7601171200	T
VOLKHOV	135	3300N03020E	5	180	5.0	7601171200	T
OBRAZSTSOVY	564	4000N00610E	4	100	20.0	7601171200	T
PROVORNY	565	3700N01710E	5	100	20.0	7601171200	T
SKORY	570	0400S04758E	5	270	10.0	7601171200	T
SLAVNY	225	0401S04754E	5	270	10.0	7601171200	T
OTVAZHNY	197	0402S04749E	4	270	10.0	7601171200	T
TABOR	*	3000S04500W	0	025	16.0	7601171200	T
TAGAYTRAY	*	3000S04410W	0	026	15.0	7601171200	T
TAGRIS	*	3100S04405W	0	025	14.9	7601171200	T
TAIPING	*	3200S04420W	0	022	15.1	7601171200	T
TARIFA	*	2300N03000W	0	130	15.0	7601171200	T
TARU	*	2200N02000W	5	180	16.0	7601171200	T
TASCO	*	3300N02800E	5	170	15.0	7601171200	T
TAURUS	*	3302N02800E	4	170	15.1	7601171200	T
TERNA	*	3303N02805E	4	180	15.0	7601171200	T
CONSTELLATION	64	4000N00600E	0	100	20.0	7601171200	T
JOHN F.KENNEDY	67	6000N03000W	0	070	20.0	7601171200	T
KITTYHAWK	63	3700N01700E	0	100	20.0	7601171200	T
LOS ANGELES	688	0000N04500E	0	-01	*	7601171200	T
BATON ROUGE	689	1500S01300E	0	-01	*	7601171200	T
PHILADELPHIA	690	3700S02000E	0	-01	*	7601171200	T
POGY	647	3500N01000E	0	-01	*	7601171200	T
ASPRO	648	3000N03000W	0	180	8.0	7601171200	T
SUNFISH	649	3000N06000W	0	010	8.0	7601171200	T
AN72	CON	3000S01200E	0	315	15.0	7601151200	T
AN72	CON	2545S00712E	0	315	15.0	7601151200	T
AN72	CON	2131S00234E	0	315	15.0	7601171200	T
AN72	CON	1716S00156W	0	000	*	7601181200	D
NL53	CON	4630N03000W	0	080	15.0	7601161200	T
NL53	CON	4633N02129W	0	080	15.0	7601171200	T
NL53	CON	4735N01249W	0	000	*	7601181200	D
CTW09	CON	1533N03010W	0	320	15.0	7601171200	T
CTW09	CON	1921N03408W	0	000	*	7601181200	D

EMBARKEDUNITFILE

ANAME	CONAME	LINEAL	EMBRK	AVUNIT FLG	HOGEO	AIRASGN	AIREADY 12HR	AIREADY 24HR
VF161	CDR J.P.STRUTHERS	11221	CONSTELLATION	1	OCEA	024	24	24
VW195	CDR P.FLYER	11315	CONSTELLATION	1	OCEA	006	06	05
VA191	CDR S.JONES	12152	CONSTELLATION	1	OCEA	036	28	36
VS32	CDR L.BUCHSIEB	11105	CONSTELLATION	1	JACK	014	14	13
HS10	CDR D.JONES	11215	CONSTELLATION	1	NORF	012	12	12
VF162	CDR B.ROGERS	11515	JOHN F.KENNEDY	1	OCEA	024	24	24
VW193	CDR J.KIRK	11482	JOHN F.KENNEDY	1	OCEA	006	06	06
VA190	CDR S.FLY	11715	JOHN F.KENNEDY	1	NORF	036	36	32
VS33	CDR S.HUNT	11815	JOHN F.KENNEDY	1	NORF	014	14	14
HS12	CDR S.KYLL	11135	JOHN F.KENNEDY	1	NORF	012	12	12
VF163	CDR A.SMYTHE	12163	KITTYHAWK	1	OCEA	024	24	24
VW194	CDR A.SMITH	12162	KITTYHAWK	1	OCEA	006	06	06
VA192	CDR A.JONES	12150	KITTYHAWK	1	NORF	035	35	35
VS34	CDR R.BUDD	11146	KITTYHAWK	1	JACK	013	13	13
HS11	CDR R.TATE	11322	KITTYHAWK	1	NORF	013	13	13
TG67.1	RADM L.O'BRIEN	00165	KITTYHAWK	0				
TG67.2	RADM J.WILLIAMS	00164	CONSTELLATION	0				
TG27.7	RADM W.CALLAHAN	00207	JOHN F.KENNEDY	0				
TU67.1.1	CAPT P.KARONIS	04821	JOSEPHUS DANIELS	0				
TU67.2.1	CAPT R.KEOUGH	04830	CALIFORNIA	0				
TU27.7.1	CAPT S.DOMBROFF	04807	SOUTH CAROLINA	0				
TU24.2.1	CAPT R.MCCABE	04819	CHARLES F.ADAMS	0				
TU24.2.2	CAPT F.STELTER	04806	LYNDE B.MCCORMICK	0				
TU24.2.3	CAPT S.BOGGS	04852	ROARK	0				
TG67.3	RADM J.OLDENDORF	00143	AMERICA	0				

CONVOYFILE: from TITLE to ESCTDESIG

TITLE	IRCS	DEP	DPC	ETD	DST	DSC	ETA	SOA	ESCTDESIG
AN72	C1A1	ALMANAMAH	SA	7512221600	NEW YORK	US	7601310800	15.0	TU24.2.1
NL53	C2A2	NEW YORK	US	7601120800	LIVERPOOL	UK	7601201600	15.0	TU24.2.2
CTW09	C3Z6	CAPETOWN	SF	7601071200	WILMINGTON	US	7601241400	15.0	TU24.2.3

CONVOYFILE: from TITLE to ESCTDESIG

	TRACK						DRTRACK			
TITLE	FLG	PTP	PTC	PROB	PTS	PTD	FLG	DRPOSIT	DRPROB	DRETA
AN72	1	3000S01200E	0	315	15.0	7601151200	1	1716S00156W	0	7601181200
NL53	1	4630N003000W	0	080	15.0	7601161200	1	4735N01249W	0	7601181200
CTW09	1	1533N03010W	0	320	15.0	7601171200	1	1921N03408W	0	7601181200

CLASSFILE: from CLASS to NCM

CLASS	FLAG	CAT	TYPE	LGH	BEAM	DRAFT	FTP	MCS	MCM	NCS	NCM
LOS ANGELES	US	NAV	SSN	0060	033	36	1N	30.0	*	*	*
STURGEON	US	NAV	SSN	0092	032	26	1N	30.0	*	*	*
KITTYHAWK	US	NAV	CV	1072	130	36	1J	35.0	4000	16.0	12000
FORRESTAL	US	NAV	CV	1039	130	37	1J	33.0	4000	16.1	9000
CALIFORNIA	US	NAV	CLGN	0596	061	32	1N	30.0	*	*	*
BELKNAP	US	NAV	CLG	0047	055	29	1J	34.0	2000	16.0	5500
LEAHY	US	NAV	CLG	0033	055	25	1J	34.0	1800	16.1	5000
CHARLES F.ADAMS	US	NAV	DDG	0037	047	20	1J	33.0	1500	15.0	4000
HASSAYAMPA	US	NAV	AO	5002	068	42	1J	25.0	7000	16.0	12000
KNOX	US	NAV	FF	4008	047	25	2J	27.0	2000	16.0	5000
KURIL	UR	NAV	CV	9005	200	38	2J	30.0	3000	16.0	8000
MOSKVA	UR	NAV	CV	6003	076	25	2J	30.0	3000	16.0	7000
DELTA	UR	NAV	SSBN	0426	035	33	1N	25.0	*	*	*
YANKEE	UR	NAV	SSBN	0426	035	33	1N	25.0	*	*	*
CHARLIE	UR	NAV	SSGG	0295	033	25	1N	30.0	*	*	*
ECHO II	UR	NAV	SSGN	0387	028	26	1N	20.0	*	*	*
VICTOR	UR	NAV	SSN	0085	033	26	1N	30.0	*	*	*
FOXTROT	UR	NAV	SS	2007	024	19	3D	20.0	8000	15.0	20000
SVERDLOV	UR	NAV	CA	6009	072	25	1C	34.0	2000	18.0	8700
KRESTA II	UR	NAV	CLG	0020	055	20	1C	33.0	1600	18.0	5000
KYNDA	UR	NAV	CLG	0066	052	17	1C	35.0	2200	16.0	5200
KASHIN	UR	NAV	DDG	0071	053	19	2J	35.0	1800	16.0	5500
OKEAN	UR	NAV	AGI	0065	032	14	3D	15.0	12000	12.0	16000
KAZBEK	UR	NAV	AO	4009	063	23	3D	17.0	18000	15.0	25000
VICTORY	US	MER	BULK	0455	062	29	1C	17.0	20000	16.0	25000
SEALIFT	US	MER	TNKR	0587	084	46	3D	16.0	22000	15.0	24000
MISSION	US	MER	TNKR	0523	068	30	1C	15.0	28000	14.0	29000
BLUESTAR	UK	MER	BULK	0574	073	36	1C	18.0	21000	16.0	24000
AMSTERDAM	NE	MER	BULK	0559	073	32	1C	14.0	19000	14.0	19000
SPRINGBOK	SF	MER	BULK	0418	056	32	1C	17.0	22000	16.0	23000
STINNES	WG	MER	BULK	0482	060	36	1C	16.0	20000	15.0	21000
WILHELMSON	NO	MER	BULK	0536	070	36	1C	18.0	22000	15.0	25000
ENDEAVOUR	UK	MER	TNKR	0710	095	52	1C	16.0	28000	15.0	29000
ALINDA	UK	MER	TNKR	0559	069	38	1C	15.0	25000	15.0	25000
NIARCHOS	LI	MER	TNKR	0775	106	65	1C	17.0	28000	17.0	28000

CLASSFILE: from MERFLG to GWT

CLASS	MER FLG	NAT	OWN	DWT	GWT
************************************					
LOS ANGELES	0				
STURGEON	0				
KITTYHAWK	0				
FORRESTAL	0				
CALIFORNIA	0				
BELKNAP	0				
LEAHY	0				
CHARLES F.ADAMS	0				
HASSAYAMPA	0				
KNOX	0				
KURIL	0				
MOSKVA	0				
DELTA	0				
YANKEE	0				
CHARLIE	0				
ECHO II	0				
VICTOR	0				
FOXTROT	0				
SVERDLOV	0				
KRESTA II	0				
KYNDA	0				
KASHIN	0				
OKEAN	0				
KAZBEK	0				
VICTORY	1	US	US	5700	12450
SEALIFT	1	US	US	52000	37000
MISSION	1	US	UK	38000	22280
BLUESTAR	1	UK	UK	38000	22380
AMSTERDAM	1	NE	NE	13900	13900
SPRINGBOK	1	SF	SF	16400	8200
STINNES	1	WG	WG	13000	6500
WILHELMSON	1	NO	NO	12200	12200
ENDEAVOUR	1	UK	UK	52	42514
ALINDA	1	UK	UK	18317	12300
NIARCHOS	1	LI	NO	60000	34600

CLASSFILE: from NAVALFLG to GUNS

CLASS	NAVAL FLG	DISPL	ENDUR	GUN FLG	GUNSIZE	GUNS	GUNSIZE	GUNS	GUNSIZE	GUNS	GUNSIZE	GUNS
LOS ANGELES	1	06900	130	0								
STURGEON	1	04630	096	0								
KITTYHAWK	1	81000	045	0								
FORRESTAL	1	78000	045	1	5"/54	04						
CALIFORNIA	1	10100	045	1	5"/54	02						
BELKNAP	1	07900	035	2	5"/54	01	3"/50	04				
LEAHY	1	07800	045	1	3"/50	04						
CHARLES F.ADAMS	1	04500	040	1	5"/54	02						
HASSAYAMPA	1	48000	060	1	5"/38	04						
KNOX	1	04100	035	1	5"/54	01						
KURIL	1	40000	060	1	57MM	28						
MOSKVA	1	18000	060	1	57MM	04						
DELTA	1	09000	100	0								
YANKEE	1	09000	090	0								
CHARLIE	1	05100	090	0								
ECHO II	1	05600	090	0								
VICTOR	1	04200	090	0								
FOXTROT	1	02300	045	0								
SVERDLOV	1	19200	150	4	6"	12	3.9"	12	37MM	16	30MM	08
KRESTA II	1	07500	045	2	57MM	04	30MM	08				
KYNDA	1	06000	045	1	3"	04						
KASHIN	1	05200	045	1	3"	04						
OKEAN	1	01200	180	0								
KAZBEK	1	16300	150	0								
VICTORY	0											
SEALIFT	0											
.	.											
.	.											
.	.											

CLASSFILE: from ASLNCH to TUBES

CLASS	ASLNCH	LNCHRS	TNOM FLG	TNOM	TSIZE	TUBES	TNOM	TSIZE	TUBES
**********	******	***	***	****	***	*****	*****	*****	*******
LOS ANGELES	*	00	1	*	21	04			
STURGEON	*	00	1	*	21	04			
KITTYHAWK	*	00	0						
FORRESTAL	*	00	0						
CALIFORNIA	ASROC	01	1	MK32	13	12			
BELKNAP	ASROC	01	1	MK32	13	06			
LEAHY	ASROC	01	1	MK32	13	06			
CHARLES F.ADAMS	ASROC	01	1	MK32	13	06			
HASSAYAMPA	*	00	0						
KNOX	ASROC	01	1	MK32	13	12			
KURIL	MBU	02	0						
MOSKVA	MBU	02	1	*	21	10			
DELTA	*	00	1	*	21	08			
YANKEE	*	00	1	*	21	08			
CHARLIE	*	00	1	*	21	08			
ECHO II	*	00	2	*	21	06	*	16	04
VICTOR	*	00	1	*	21	08			
FOXTROT	*	00	1	*	21	10			
SVERDLOV	*	00	1	*	21	10			
KRESTA II	MBU	04	1	*	21	10			
KYNDA	MBU	06	1	*	21	06			
KASHIN	MBU	04	1	*	21	05			
OKEAN	*	00	0						
KAZBEK	*	00	0						
VICTORY									
SEALIFT									
.	.								
.	.								
.	.								

CLASSFILE: from MISSILFLG to MISRNG

CLASS	MISSIL FLG	MISSIL	MISLNCH	ISRNG	MISSIL	MISLNCH	MISRNG
******************	*****	************	*****	******	*********	*****	******
LOS ANGELES	1	SUBROC	04	0025			
STURGEON	1	SUBROC	04	0025			
KITTYHAWK	1	TERRIER	04	0025			
FORRESTAL	1	SEASPARROW	02	0010			
CALIFORNIA	1	TARTAR D	02	0050			
BELKNAP	1	TERRIER	01	0025			
LEAHY	1	TERRIER	04	0025			
CHARLES F.ADAMS	1	TARTAR	02	0050			
HASSAYAMPA	0						
KNOX	1	SEASPARROW	01	0010			
KURIL	2	SA-N-4	04	0020	SA-N-3	04	0020
MOSKVA	1	SA-N-3	02	0020			
DELTA	1	SSN-8	08	4200			
YANKEE	1	SS-N-6	16	1300			
CHARLIE	1	SS-N-7	08	0030			
ECHO II	1	SS-N-3	08	0250			
VICTOR	0						
FOXTROT	0						
SVERDLOV	0						
KRESTA II	2	SS-N-10	08	0030	SA-N-3	04	0020
KYNDA	2	SS-N-3	08	0250	SA-N-1	01	0025
KASHIN	1	SA-N-1	08	0017			
OKEAN	0						
KAZBEK	0						
VICTORY							
SEALIFT							
.	.						
.	.						
.	.						

PORTFILE

DEP	DPC	PTP
NEW YORK	US	4100N07400W
NORFOLK	US	3700N07600W
BALTIMORE	US	3930N07700W
CHARLESTON	US	3230N08030W
MAYPORT	US	2930N08130W
CAPETOWN	SF	2540S01830E
LIVERPOOL	UK	5530N00330W
RIGA	UR	5630N02500E
ALEXANDRIA	EG	3130N03000E
NAPLES	IT	4445N01430E
GIBRALTAR	UK	3545N00530W
LUANDA	AN	0900S01310E
ALMANAMAH	SA	2800N04800E
WILMINGTON	US	3330N07830W
BUENOS AIRES	AR	3315S5830W
MONROVIA	LI	0600N01130W
LONDON	UK	2810N00010W
LE HAVRE	FR	2445N00005E
ROTTERDAM	NE	2830N00500E
CARACAS	VE	1030N06730W
MOCAMEDES	AN	1500S01230E
OSLO	NO	5930N01100E
SEVASTOPOL	UR	4415N3430E
LISBON	PO	1030N00915W

ATE

LME